

# E<sup>2</sup>M<sup>2</sup>: Energy Efficient Mobility Management in Dense Small Cells with Mobile Edge Computing

Jie Xu\*, Yuxuan Sun<sup>†</sup>, Lixing Chen\*, Sheng Zhou<sup>†</sup>

\*Department of Electrical and Computer Engineering, University of Miami, USA

<sup>†</sup>Department of Electronic Engineering, Tsinghua University, China

**Abstract**—Merging mobile edge computing with the dense deployment of small cell base stations promises enormous benefits such as a real proximity, ultra-low latency access to cloud functionalities. However, the envisioned integration creates many new challenges and one of the most significant is mobility management, which is becoming a key bottleneck to the overall system performance. Simply applying existing solutions leads to poor performance due to the highly overlapped coverage areas of multiple base stations in the proximity of the user and the co-provisioning of radio access and computing services. In this paper, we develop a novel user-centric mobility management scheme, leveraging Lyapunov optimization and multi-armed bandits theories, in order to maximize the edge computation performance for the user while keeping the user’s communication energy consumption below a constraint. The proposed scheme effectively handles the uncertainties present at multiple levels in the system and provides both short-term and long-term performance guarantee. Simulation results show that our proposed scheme can significantly improve the computation performance (compared to state of the art) while satisfying the communication energy constraint.

## I. INTRODUCTION

Mobile networks are experiencing a paradigm shift. A multitude of new technologies are being developed to accommodate the surging data demand from mobile users. Among these technologies, Ultra Dense Networking (UDN) [1][2] and Mobile Edge Computing (MEC) (a.k.a. Fog Computing) [3][4] are considered as key building blocks for the next generation mobile network (5G). UDN increases the network capacity through the ultra-dense deployment of small cell Base Stations (BSs), which is viewed as the key technology to realize the so-called 1000x capacity challenge [5]. MEC provides access to cloud-like computing and storage resources at the edge of the mobile network, creating significant benefits such as ultra-low latency and precise location awareness, which are necessary for emerging applications such as Tactile Internet, Augmented Reality, Connected Cars and the Internet of Things. In many ways, UDN may provide the strongest case for MEC since small cell BSs are inherently edge-oriented. It is envisioned that small cell BSs endowed with cloud functionalities will be a major form of MEC service provision nodes [6].

Although there are increasingly many works studying UDN and MEC, they are mostly separate efforts. However, significant new challenges are created by the envisioned integration of UDN and MEC, despite the enormous potential benefits. A key bottleneck to the overall system performance in this new paradigm is mobility management, which is the fundamental

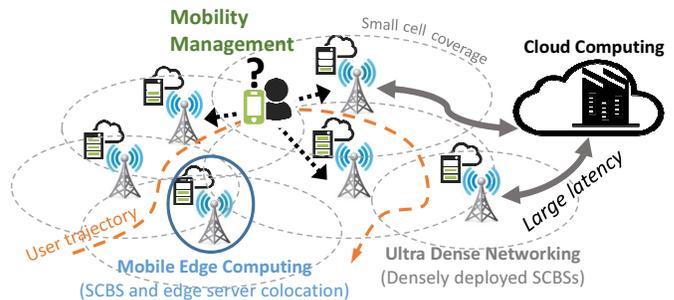


Fig. 1. Mobility management in small cell-based MEC.

function of tracking mobile User Equipments (UEs) and associating them with appropriate BSs, thereby enabling mobile service to be delivered. Traditionally, mobility management was designed for macro cellular networks with infrequent handovers between cells, and providing radio access service only. Merging UDN and MEC drastically complicates the problem. Simply applying existing solutions leads to poor mobility management due to the highly overlapped coverage areas of multiple BSs in the proximity and the co-provisioning of radio access and computing services. In particular, what impedes efficient mobility management is the tremendous uncertainty at multiple levels:

- **Present uncertainties.** A key challenge is the unavailability of accurate information of candidate BSs in the proximity. Had the system known a priori which BSs offers the best performance (either throughput, energy consumption or computation latency), the mobility management decision making could be much easier, thus avoiding frequent handovers which lead to energy inefficiency.
- **Future uncertainties.** A perhaps even bigger challenge is that the long-term energy budget couples the mobility management decisions across time, and yet the decisions have to be made without foreseeing the future (i.e. the future locations, channel conditions, available edge computing resources etc.)

Figure 1 illustrates the mobility management problem in small cell-based MEC. In this paper, we develop a novel user-centric mobility management scheme, called E<sup>2</sup>M<sup>2</sup> (Energy-Efficient Mobility Management), that (i) smartly learns the best BSs to connect and makes proper handover decisions in an online fashion, and (ii) satisfies both short-term and

long-term performance requirement. This is mainly achieved by leveraging and extending Lyapunov Optimization [7] and Multi-armed Bandits [8]. In particular, we provide strong performance guarantee by employing our scheme: (1) the computing performance is within a bounded deviation of the optimal performance that can be achieved by an imaginary oracle that knows all information in advance; (2) the long-term energy consumption constraint is approximately satisfied within a bounded deviation.

## II. RELATED WORK

Mobile edge computing has received an increasing amount of attentions in recent years. A central theme of many prior studies is offloading policies, i.e. what/when/how to offload a user's workload from its device to the edge system or cloud (see [9][10] and references therein). Our paper does not study offloading policies and can work in conjunction with any offloading policy. There is also extensive work on dense small cell networks, mainly focusing on radio resource management and interference mitigation (see [11][12] and references therein). Our paper is also orthogonal to this literature and does not impose any assumption on how the network is planned or how the radio resource is allocated among the cells (hence we allow inter-cell interference in our model).

Much of the mobility management research has been based on optimization theory [13][14][15][16]. For instance, a utility maximization problem is formulated for the optimal user association [13]. A handover policy is proposed in [16] in which low energy efficiency from the serving BS triggers a handover. These solutions have been proved effective for less-densified heterogeneous networks, but they may perform poorly when the network density becomes high as frequent handovers may occur. To address this challenge, a learning-based mobility management scheme is proposed in [17] based on the multi-armed bandits technique [8]. This scheme is user-centric, which has been an emerging trend of designing mobility management, particularly for the future 5G standard [18]. However, all these works merely consider the radio access aspect of BSs. Endowing BSs with MEC capabilities requires new mobility management solutions.

To deal with future uncertainties, we leverage Lyapunov optimization [7], which has wide applications in communication networks and queueing systems. Our scheme builds upon yet extends Lyapunov optimization by integrating Multi-armed Bandits [8] to solve the optimization task in each time period due to the lack of accurate system information.

## III. SYSTEM MODEL

### A. Network Model

We consider an ultra-dense cellular network with  $N$  Base Stations (BS) (which can be macro BSs or small cell BSs), indexed by  $\mathcal{N} = \{1, 2, \dots, N\}$ . The network is divided into  $M$  disjoint regions, indexed by  $\mathcal{M} = \{1, 2, \dots, M\}$ . Each region  $m \in \mathcal{M}$  is covered by a subset of BSs, denoted by  $\mathcal{A}(m) \subseteq \mathcal{N}$ . We consider a representative mobile user that is moving in the network for  $T$  time periods. In period  $t$ , the location

of the user is denoted by  $L^t \in \mathcal{M}$ . Therefore,  $\mathcal{L} = L^1 \rightarrow L^2 \rightarrow \dots \rightarrow L^T$  is the trajectory of the user. This trajectory can be generated by any mobility model, such as the random waypoint model, and we do not require any prior knowledge about the trajectory.

At each location  $L^t$  in time period  $t$ , the mobility management system makes decisions on which BS the user is served by, among the discoverable BSs  $\mathcal{A}(L^t)$  in the vicinity. The mobility decision is traditionally made at the BS (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 [18]. In this work, we consider user-centric mobility management and let the user make mobility decisions.

### B. Workload and Service Model

In each period  $t$ , the user has computation workloads with arrival rate  $\lambda^t \in [0, \lambda_{\max}]$  that needs to be offloaded to the mobile edge server for processing, where  $\lambda_{\max}$  is the maximum possible arrival rate. We assume that each workload has an equal size  $\gamma$ . As assumed in prior work,  $\lambda^t$  is available at the beginning of each time period  $t$ . We focus on delay-sensitive interactive workloads, which are the main application scenario of MEC.

Each BS is co-located with an edge server. The maximum service rate of BS  $n$  in time period  $t$  is denoted by  $s^t(n)$ . The total workload offloaded to BS  $n$  by other users is denoted by  $\mu^t(n)$ , which we refer to as background workload. By offloading  $\lambda^t$  to a BS  $n \in \mathcal{A}(L^t)$ , the delay cost incurred to the representative user can be captured by a general function  $d(\lambda^t, \mu^t(n), s^t(n))$ . As a concrete example, we model the service process at each edge server as a M/M/1 queue and the average response time (multiplied by the arrival rate) to represent the delay cost. Specifically, the delay cost in period  $t$  can be written as [19]

$$d(\lambda^t, \mu^t(n), s^t(n)) = \frac{\lambda^t}{s(n) - (\mu^t(n) + \lambda^t)} \quad (1)$$

in which we ignore the transmission delay which is typically small since the BS is in close proximity. While the M/M/1 queueing model may not capture the exact response time in practice, it has been widely used as an analytic vehicle to provide a reasonable approximation for the actual service process. It is also possible that the workload is further offloaded a remote cloud or distributed among a set of edge servers. Our framework is able to capture this scenario by using a proper delay cost function  $d(\cdot)$  that absorbs this additional delay. For simplicity, this paper focuses on processing of computation workload on local edge servers.

### C. Channel and Power Consumption Model

Since the considered period is relatively long, we consider only slow fading for the wireless channel for uploading the workload to the edge server. Let  $H^t(n, m)$  denote the wireless channel state between region  $m$  and BS  $n \in \mathcal{A}_m$  in period  $t$ .

Given the transmission power  $P_{tx}$  of the user, the maximum achievable uplink transmission rate is given by the Shannon's theorem,

$$r(H^t(n, m)) = W \log_2 \left( 1 + \frac{P_{tx} H^t(n, m)}{\sigma^2 + I^t(n)} \right) \quad (2)$$

where  $W$  is the channel bandwidth and  $\sigma^2$  is the noise power and  $I^t(n)$  is the possible inter-cell interference in time period  $t$ . The required transmission energy for uploading workload with arrival rate  $\lambda^t$  to BS  $n$  in a period is therefore

$$e(\lambda^t, H^t(n, L^t)) = \frac{P_{tx} \lambda^t \gamma}{W \log_2 \left( 1 + \frac{P_{tx} H^t(n, L^t)}{\sigma^2 + I^t(n)} \right)} \quad (3)$$

#### IV. PROBLEM FORMULATION

We first formulate the problem by assuming that the user has complete information of the system for all time periods  $t = 0, 1, \dots, T-1$  in advance at time  $t = 0$ . Therefore, the user essentially is solving an offline optimization problem. However, in the real system, information in period  $t$  can only be known in period  $t$ . Therefore, we will subsequently develop online algorithms and show that they are efficient compared to the optimal offline algorithm. For the online algorithms, we will consider two practical scenarios depending on the availability of information. The information in each period  $t$  can be classified into two categories:

- **User-side State Information:** The user's location  $L^t$ , the available candidate BSs  $\mathcal{A}^t$  and the workload rate  $\lambda^t$ .
- **BS-side State Information:** For each  $n \in \mathcal{A}^t$ , its background workload  $\mu^t(n)$ , its maximum service rate  $s^t(n)$  and the uplink channel condition  $H^t(n, L^t)$ .

The first deployment scenario assumes that the user has both the user-side information and the BS-side information. The second deployment scenario assumes that the user has only the user-side information.

##### A. Offline Problem Formulation

The user performs mobility management to maximize its computing performance (i.e. to minimize the delay cost) subject to an energy consumption budget (e.g. since battery is limited). Formally, the problem is as follows

$$\mathbf{P1:} \quad \min_{a^0, \dots, a^{T-1}} \frac{1}{T} \sum_{t=0}^{T-1} d(\lambda^t, \mu^t(a^t), s^t(a^t)) \quad (4)$$

$$\text{s.t.} \quad \sum_{t=0}^{T-1} e(\lambda^t, H^t(a^t, L^t)) \leq \alpha B \quad (5)$$

$$r(H^t(a^t, L^t)) \geq r_{\min}, \forall t \quad (6)$$

$$d(\lambda^t, \mu^t(a^t), s^t(a^t)) \leq d_{\max}, \forall t \quad (7)$$

$$a^t \in \mathcal{A}(L^t) \quad (8)$$

The objective is to minimize the average delay cost, which is the key performance metric for MEC applications which require real-time interaction, by deciding which BS the user should be served by in each time period. The first constraint (5) states that the total energy consumption is limited by the

available battery during the current trip of the user, where  $\alpha \in (0, 1]$  indicates the desired capping of energy consumption relative to the total battery capacity  $B$ . The second constraint (6) requires that the uploading transmission rate meets a minimum rate  $r_{\min}$  so that the transmission delay is negligible. The third constraint (7) requires that the per-slot delay does not exceed an upper limit  $d_{\max}$  so the real-time performance is guaranteed in the worst case. The last constraint (8) states that the candidate BSs are those that cover  $L^t$ .

The first major practical challenge that impedes derivation of the optimal solution to **P1** is the lack of future information: optimally solving **P1** requires complete offline information (including the user-side and the BS-side information) over the entire trip periods that is extremely difficult, if not impossible, to accurately predict in advance. Furthermore, **P1** belongs to integer nonlinear programming and is difficult to solve, even if the long-term future information is accurately known a priori. Thus, these challenges demand an online approach that can efficiently make mobility management decisions without foreseeing the far future.

##### B. J-Step Lookahead Algorithm

Because the system state information (i.e. the location, the workload, the channel state etc.) can follow an arbitrary sample path, we use the  $J$ -step lookahead algorithm as an **benchmark** to evaluate our online algorithms. Specifically, we divide the entire trip duration into  $R \geq 1$  frames, each having  $J \geq 1$  time periods such that  $T = RJ$ . For the  $r$ -th frame, we define  $D_r^*$  as the optimal delay cost associated with the following static optimization problem.

$$\mathbf{P2:} \quad \min_{a^{rJ}, \dots, a^{(r+1)J-1}} \frac{1}{J} \sum_{t=rJ}^{(r+1)J-1} d(\lambda^t, \mu^t(a^t), s^t(a^t)) \quad (9)$$

$$\text{s.t.} \quad \sum_{t=rJ}^{(r+1)J-1} e(\lambda^t, H^t(a^t, L^t)) \leq \frac{\alpha B}{R} \quad (10)$$

$$\text{constraints (6), (7), (8)} \quad (11)$$

The value  $D_r^*$  thus represents the optimal empirical average delay cost for frame  $r$  over all policies that have full knowledge of the future system state information over the frames and that satisfy the constraints. We assume that for every frame, there exists at least one sequence of (possibly randomized) mobility management decisions that satisfy the constraints of **P2**. Let  $a^{rJ,*}, \dots, a^{(r+1)J-1,*}$  denote the sequence of the optimal mobility management decisions that achieve  $D_r^*$ . The minimum long-term average delay cost achieved by the oracle's optimal  $J$ -step lookahead algorithm is thus given by  $D^* = \frac{1}{R} \sum_{r=0}^{R-1} D_r^*$ .

#### V. ONLINE MOBILITY MANAGEMENT

In this section, we develop online mobility management algorithms that do not require future system state information. We consider two deployment scenarios depending on whether the user has the exact BS-side state information.

### A. $E^2M^2$ Algorithm with Full State Information

We first consider the case in which the user has full state information in each period. A significant challenge of directly solving **P1** is that the long-term energy budget couples the mobility management decisions across different time periods: using more energy at the current time will potentially reduce the energy budget available for future uses, and yet the decisions have to be made without foreseeing the future. To address this challenge, we leverage Lyapunov optimization and construct a virtual energy deficit queue to guide the mobility management decisions. Specifically, let  $q(0) = 0$ , the dynamics of the energy deficit queue evolves as

$$q(t+1) = \{q(t) + e(\lambda^t, H^t(a^t, L^t)) - \alpha B/T\}^+ \quad (12)$$

The virtual queue length  $q(t)$  indicates how far the current energy usage deviates from the battery energy budget. The algorithm is shown in Algorithm 1.

$E^2M^2$  is an online algorithm because it requires only the currently available information as the inputs. We use  $V_0, V_1, \dots, V_{R-1}$  to denote a sequence of positive control parameters to dynamically adjust the tradeoff between delay cost minimization and energy consumption over the  $R$  frames, each having  $J$  periods. Lines 2 - 4 reset the energy deficit virtual queue at the beginning of each frame. Line 5 defines an online optimization problem **P3** to decide the mobility management decisions. The optimization problem aims to minimize a weighted sum of the delay cost and energy consumption where the weight depends on the current energy deficit queue length. A large weight will be placed on the energy consumption if the current energy deficit is large. The energy deficit queue maintained without foreseeing the future guides the mobility management decisions towards meeting the battery energy constraint, thereby enabling online decisions.

---

#### Algorithm 1 $E^2M^2$ with Full State Information (FSI)

---

- 1: **Input:**  $L^t, \mathcal{A}^t, \lambda^t$ , and  $\forall n \in \mathcal{A}^t, \mu^t(n), s(n), H^t(n, L^t)$  at the beginning of each  $t$ .
  - 2: **if**  $t = rJ, \forall r = 0, 1, \dots, R - 1$  **then**
  - 3:      $q(t) \leftarrow 0$  and  $V \leftarrow V_r$
  - 4: **end if**
  - 5: Choose  $a^t$  subject to (6), (7), (8) to minimize
 
$$(\mathbf{P3}) \quad V \cdot d(\lambda^t, \mu^t(a^t), s(a^t)) + q(t) \cdot e(\lambda^t, H^t(a^t, L^t))$$
  - 6: Update  $q(t)$  according to (12).
- 

### B. $E^2M^2$ Algorithm with Partial State Information

In many deployment scenarios, the user knows only the user-side information but not the BS-side information. This creates a major difficulty in solving **P3** exactly in each period  $t$ . For notational convenience, we define  $d(n) = d(\lambda^t, \mu^t(n), s(n))$  and  $e(n) = e(\lambda^t, H^t(n, L^t))$  for all  $n \in \mathcal{A}(L^t)$ , and  $Z(n) = V \cdot d(n) + q(t) \cdot e(n)$ . In each period  $t$ , the optimal BS for serving the user is thus

$$a^{t,*} = \arg \min_n \{Z(n)\} \quad (13)$$

Since  $\mu^t(n), s^t(n)$  and  $H^t(n, L^t)$  for any  $n \in \mathcal{A}(L^t)$  are unknown by the user, the values of  $d(n)$  and  $e(n)$  and hence  $Z(n)$  cannot be evaluated exactly. In this section, we augment our  $E^2M^2$  algorithm with an online learning algorithm that learns  $a^{t,*}$  without requiring the exact BS-side information.

In order to learn the optimal mobility management decision, we divide each period  $t$  into  $K$  slots. In each time slot within a time period, the learning algorithm connects the user with one BS  $a_k^t \in \mathcal{A}(L^t)$ . At the end of the time slot  $k$ , the user observes the delay  $\tilde{d}_k^t$  and energy consumption  $\tilde{e}_k^t$ , which serve as feedback information to guide the learning of the optimal BS. Let  $z_k^t = V \cdot \tilde{d}_k^t + q(t) \cdot \tilde{e}_k^t$ . If  $z_k^t$  were exactly  $\frac{1}{K}Z(a^t)$ , then learning the optimal  $a^{t,*}$  would be simple: use each  $n \in \mathcal{A}(L^t)$  for one slot in a round-robin fashion and keep connecting to the one that minimizes  $z_k^t$  for the remaining  $K - |\mathcal{A}(L^t)|$  slots. However, due to the variance in workload arrivals and channel states,  $z_k^t$  is only a noisy version of  $\frac{1}{K}Z(a^t)$ . As a result, the above simple learning algorithm can be very suboptimal since we may get trapped in a BS that results in a large  $Z(n)$ . In fact, this problem is a classic sequential decision making problem that involves a tradeoff between exploration and exploitation – the learning algorithm needs to explore the different BSs to learn good estimates of  $Z(n), \forall n$  while at the same time trying to connect to the optimal BS as long as possible. This problem is extensively studied under the multi-armed bandits framework and many learning algorithms have been developed with performance guarantee. In this paper, we augment our  $E^2M^2$  algorithm with the widely adopted UCB1 algorithm [8] to learn the optimal BS under uncertainty. UCB1 is an index-based algorithm which, in each slot  $k$ , selects the BS with the largest index. The index for BS  $n$  is an upper confidence bound on the empirical estimate of  $Z(n)$ . Nevertheless, other learning algorithms can also be incorporated in our framework.

The algorithm is shown in Algorithm 2. The major difference from Algorithm 1 is that instead of solving **P3** exactly, we use the UCB1 algorithm to learn the optimal BS to minimize the objective in **P3**, which is reflected in Lines 5 through 13.

---

#### Algorithm 2 $E^2M^2$ with Partial State Information (PSI)

---

- 1: **Input:**  $L^t, \mathcal{A}^t, \lambda^t$  at the beginning of each  $t$ .
  - 2: **if**  $t = rJ, \forall r = 0, 1, \dots, R - 1$  **then**
  - 3:      $q(t) \leftarrow 0$  and  $V \leftarrow V_r$
  - 4: **end if**
  - 5: **for** Each  $n \in \mathcal{A}(L^t)$  **do**
  - 6:      $\bar{z}^t(n) \leftarrow 0, \theta^t(n) \leftarrow 0$
  - 7: **end for**
  - 8: **for**  $k = 1, \dots, K$  **do**  $\triangleright$  UCB1 Learning
  - 9:     Connect to  $n_k^* = \arg \min_n \bar{z}^t(n) - \sqrt{\frac{\beta \ln k}{\theta^t(n)}}$
  - 10:     Observe  $\tilde{d}_k^t$  and  $\tilde{e}_k^t$
  - 11:     Update  $\bar{z}^t(n_k^*) \leftarrow \frac{\theta(n_k^*)\bar{z}^t(n_k^*) + V\tilde{d}_k^t + q(t)\tilde{e}_k^t}{\theta^t(n_k^*) + 1}$
  - 12:     Update  $\theta^t(n_k^*) \leftarrow \theta^t(n_k^*) + 1$
  - 13: **end for**
  - 14: Update  $q(t)$  according to (12).
-

## VI. PERFORMANCE ANALYSIS

We only characterize the performance of the  $E^2M^2$  with PSI because  $E^2M^2$  with FSI can be considered as a special case.

First, we characterize the performance of the UCB1 algorithm, which is used to approximately solve **P3**. Let  $\tilde{Z}^t = \sum_{k=1}^K \tilde{z}_k^t$  be the empirical total weighted sum of delay cost and energy consumption obtained by running the UCB1 algorithm. Let  $Z^{t,*} = Z(a^{t,*})$  be the optimal value with full information.

**Proposition 1.** *If each period is divided into  $K$  slots and let  $\beta = 2(z_{max}^t)^2$ , then*

$$\mathbb{E}\tilde{Z}^t - Z^{t,*} \leq z_{max}^t \left[ 8 \sum_{n \neq a^{t,*}} \left( \frac{\ln K}{\delta^t(n)} \right) + \left( 1 + \frac{\pi^2}{3} \right) \sum_{n \neq a^{t,*}} \delta^t(n) \right] \quad (14)$$

where  $\delta^t(n) = (Z^t(n) - Z^{t,*})/K$  and  $z_{max}^t = \sup z_k^t$ .

*Proof.* This is based on the regret analysis of UCB1 [8].  $\square$

Proposition 1 states that our learning algorithm approximately solves **P3** with a bounded deviation. Therefore, there exists a constant  $C$  such that  $\tilde{Z}^t - Z^{t,*} \leq C, \forall t$ .

Next, we characterize the performance of  $E^2M^2$  with PSI.

**Theorem 1.** *For any  $J \in \mathbb{Z}_+$  and  $R \in \mathbb{Z}_+$  such that  $T = RJ$ , the following statements hold.*

(1) *The average delay cost achieved by  $E^2M^2$  with PSI satisfies*

$$d^* \leq \frac{1}{R} \sum_{r=0}^{R-1} D_r^* + \frac{U(J+1) + C}{R} \sum_{r=0}^{R-1} \frac{1}{V_r} \quad (15)$$

where  $U$  is a finite constant.

(2) *The energy consumption constraint is approximately satisfied with a bounded deviation:*

$$\sum_{t=0}^{T-1} e(\lambda^t, H^t(a^t, L^t)) \leq \alpha B + \sum_{r=0}^{R-1} \sqrt{2UJ(J+1) + CJ + V_r JD_r^*} \quad (16)$$

*Proof.* See Appendix.  $\square$

Theorem 1 proves a strong performance guarantee of our mobility management scheme: the computing performance in terms of delay cost is within a bounded deviation of the optimal performance that can be achieved by an imaginary oracle that knows future information and has the computational power to solve the offline optimization problem, while the long-term energy consumption constraint is approximately satisfied within a bounded deviation.

## VII. SIMULATIONS

In this section, we carry out simulation studies to validate our analysis and evaluate the performance of the proposed mobility management scheme.

### A. Simulation Setup

We simulate a  $1000\text{m} \times 1000\text{m}$  square area in which 25 BSs are deployed on a regular grid network. The distance between two adjacent BSs is 160m. We used an adjusted random walk model to generate user trajectories. The mobility model is adjusted to avoid frequent moving back and forth in order to better capture the real world scenario. Since BSs are densely deployed, the user is able to observe multiple candidate BSs and associate to any one of them. The association radius is set to be 250m. Each time period is 5 minutes. The arrival rate of the user computational workload is set to be  $\lambda^t \in [0, 12]$  per period. The edge server service rate is  $s(n) = 50$  per period for all  $n$  and the background workload is  $\mu(n) \in [0, 40]$  per period. The wireless channel state  $H^t$  is modeled using a pathloss model  $P = 25.3 + 37.6 \times \log d$ , which is suggested in [20] for system simulations of small cells and heterogeneous networks. Other default parameters for the simulation are: noise power  $\sigma^2 = 10^{-10}\text{W/Hz}$ , channel bandwidth  $W = 20\text{M/Hz}$ , minimum transmit rate  $r_{min} = 100\text{Mbps}$ , maximum delay  $d_{max} = 5$  sec, workload size  $\gamma = 1\text{MB}$ , user transmit power  $P_{tx} = 0.1\text{W}$ .

### B. Results

1) *Performance evaluation:* We compare the performance of  $E^2M^2$  with three benchmark schemes:

- **Delay Optimal:** the user always associates with the BS with the lowest delay cost and disregards the energy consumption constraint.
- **Energy Optimal:** the user always associates with the BS with the best channel condition without considering the computing performance
- **J-step Lookahead:** this is the offline algorithm described in Section IV.B, we set  $R = 250$  and  $J = 4$ . Note that solving the offline problem is extremely computationally complex even if the future information is known.

Figure 2 shows the delay and energy performance of  $E^2M^2$  with FSI and PSI, and the benchmark schemes. As can be seen, **Delay Optimal** achieves the best delay performance at the cost of violating the energy budget constraint. **Energy Optimal** uses energy conservatively by always connecting to the BSs with the best channel condition. However, the resulting delay cost is significantly higher than other schemes. Both **J-step Lookahead** and our two  $E^2M^2$  algorithms satisfy the energy consumption constraint while keeping the delay cost low. Among them, **J-step Lookahead** is the best as expected. Despite that the proposed  $E^2M^2$  algorithms require no future information and computationally simple, they achieve comparable performance to **J-step Lookahead**. Moreover,  $E^2M^2$  with PSI is just slightly worse than  $E^2M^2$  with FSI.

### C. $E^2M^2$ with PSI

We now take a further look at  $E^2M^2$  with PSI and illustrate the performance of UCB1. Figure 3(a) shows that by augmenting  $E^2M^2$  with UCB1, the BS-side information can be learned very quickly and the convergence of the average weighted delay and energy cost to the optimal solution of **P3** is fast.

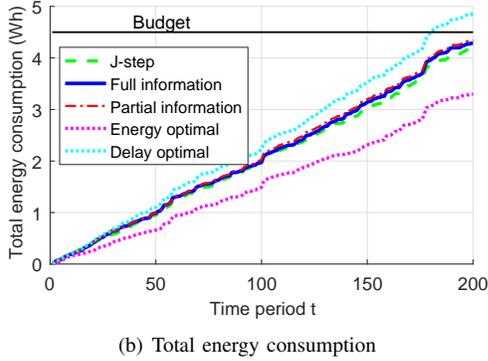
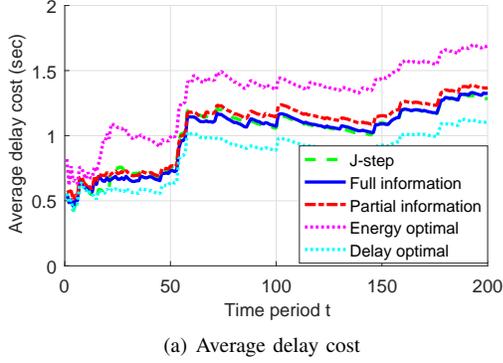


Fig. 2. Performance of  $E^2M^2$  ( $V = 0.01$ ,  $\alpha B = 4.5$ ,  $K = 100$ )

Figure 3(b) shows that the number of switchings between BSs is small during the learning process, even when the number of time slots in a period is large. This is much desired as frequent handover is avoided.

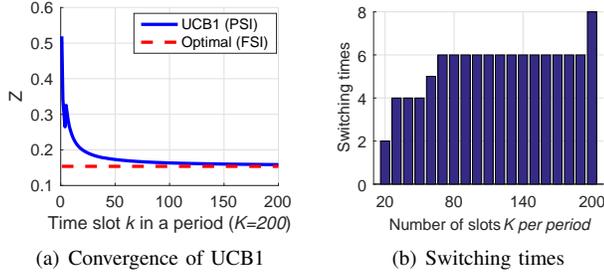


Fig. 3. Dynamics of UCB1

#### D. Impact of energy budget

Figure 4 shows the impact of the energy budget. When the energy budget is large,  $E^2M^2$  achieves the optimal delay performance since the energy constraint is always satisfied. However, it is possible there is no feasible solution when the energy budget is too low, in which case the energy constraint is violated. In between,  $E^2M^2$  makes trade-off between delay cost and energy consumption.

### VIII. CONCLUSIONS

In this paper, we studied the mobility management problem for the next generation mobile networks where the integration

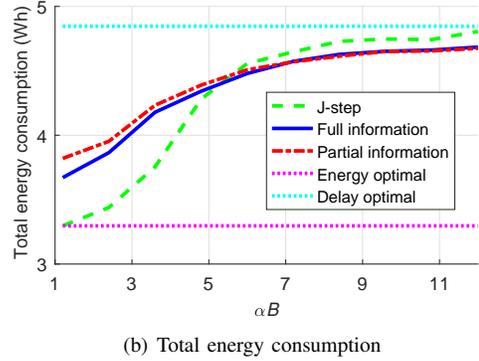
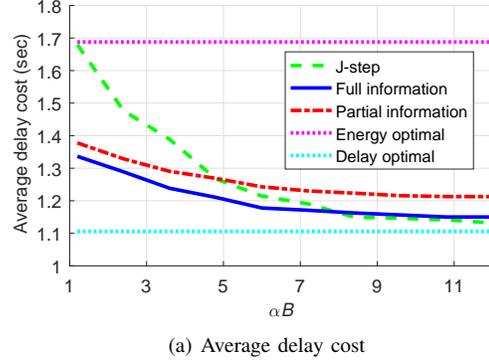


Fig. 4. Impact of battery capacity  $\alpha B$

of MEC and UDN is envisioned. We developed a novel user-centric mobility management scheme that maximizes the edge computation performance while keeping the user's communication energy consumption low. The proposed scheme works online and effectively handles present and future uncertainties in the system. Future work includes incorporating energy harvesting capabilities into mobile user device and explicitly modeling the handover cost.

### APPENDIX

For notational convenience, we define  $y(t) = e(\lambda^t, H^t(a^t, L^t)) - \alpha B/T$  and  $d(t) = d(\lambda^t, \mu^t(a^t), s(a^t))$ . First, according to the energy deficit queue dynamics, it is easy to see

$$q(t+1) - q(t) \geq y(t) \quad (17)$$

Summing the above over  $t = rJ, \dots, (r+1)J - 1$ , using the law of telescoping sums and noting  $q(rJ)$  implies

$$\sum_{t=rJ}^{(r+1)J-1} y(t) \leq q((r+1)J) \quad (18)$$

where  $q((r+1)J)$  is the queue length before reset in period  $(r+1)J$ . In what follows, we try to bound  $q((r+1)J)$ .

We define the quadratic Lyapunov function  $L(q(t)) \triangleq \frac{1}{2}q^2(t)$ . Squaring the queuing dynamics equation results in the following bound

$$\begin{aligned} q^2(t+1) &\leq (q(t) + y(t))^2 \\ &= q^2(t) + y^2(t) + 2q(t)y(t) \end{aligned} \quad (19)$$

Therefore, the 1-period Lyapunov drift  $\Delta_1(t)$  satisfies

$$\Delta_1(t) = L(q(t+1)) - L(q(t)) \leq \frac{1}{2}y^2(t) + q(t)y(t) \quad (20)$$

Now define  $U$  as a positive constant that upper bounds  $\frac{1}{2}y^2(t)$ . Such a constant exists because of the boundedness assumption. Then it can be easily shown that

$$\Delta_1(t) + V \cdot d(t) \leq U + V \cdot d(t) + q(t) \cdot y(t) \quad (21)$$

The online algorithm in line 5 of Algorithm 1 actually minimizes the upper bound on the 1-period Lyapunov drift plus a weighted delay cost shown on the right hand side of (21). Following (21), the  $J$ -period drift  $\Delta_J(rJ) \triangleq L(q((r+1)J)) - L(q(rJ))$  satisfies

$$\begin{aligned} & \Delta_J(rJ) + V \sum_{t=rJ}^{(r+1)J-1} d(t) \\ & \leq UJ + V \sum_{t=rJ}^{(r+1)J-1} d(t) + \sum_{t=rJ}^{(r+1)J-1} q(rJ) \cdot y(t) \\ & \quad + \sum_{t=rJ}^{(r+1)J-1} (q(t) - q(rJ)) \cdot y(t) \\ & \leq UJ(J+1) + V \sum_{t=rJ}^{(r+1)J-1} d(t) + \sum_{t=rJ}^{(r+1)J-1} q(rJ) \cdot y(t) \end{aligned} \quad (22)$$

where the second inequality is because  $q(t) - q(rJ) \leq (t - rJ)y_{\max}$ , and hence the last term on the right hand side satisfies

$$\sum_{t=rJ}^{(r+1)J-1} (q(t) - q(rJ)) \cdot y(t) \leq \frac{J^2}{2}y_{\max}^2 \leq J^2U \quad (23)$$

By applying  $E^2M^2$  on the left-hand side and considering the optimal  $J$ -step lookahead algorithm on the right-hand side, we obtain the following

$$\begin{aligned} & \Delta_J(rJ) + V_r \sum_{t=rJ}^{(r+1)J-1} d^*(t) \\ & \leq UJ(J+1) + V_r JD_r^* + CJ \end{aligned} \quad (24)$$

where  $d^*(t)$  is the delay cost achieved by  $E^2M^2$  at time  $t$  and  $CJ$  is due to the approximate solution of **P3**. Therefore,

$$\begin{aligned} & q((r+1)J) = \sqrt{2\Delta_J(rJ)} \\ & \leq \sqrt{2UJ(J+1) + CJ + V_r JD_r^*} \end{aligned} \quad (25)$$

Then, by (18), we have

$$\sum_{t=rJ}^{(r+1)J-1} y(t) \leq \sqrt{2UJ(J+1) + CJ + V_r JD_r^*} \quad (26)$$

By summing over  $r = 0, 1, \dots, R-1$  we prove part (2) of Theorem 1.

By dividing both sides of (24) by  $V_r$  and considering  $q(rJ) = 0$ , it follows that

$$\sum_{t=rJ}^{(r+1)J-1} d^*(t) \leq JD_r^* + \frac{UJ(J+1) + CJ}{V_r} \quad (27)$$

Thus, by summing over  $r = 0, 1, \dots, R-1$  and dividing both sides by  $RJ$ , we prove part (1) of Theorem 1.

## REFERENCES

- [1] T. Q. Quek, G. de la Roche, I. Guvenc, and M. Kountouris, *Small cell networks: Deployment, PHY techniques, and resource management*. Cambridge University Press, 2013.
- [2] I. Chih-Lin, C. Rowell, S. Han, Z. Xu, G. Li, and Z. Pan, "Toward green and soft: a 5g perspective," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 66–73, 2014.
- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.
- [4] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *arXiv preprint arXiv:1602.00484*, 2016.
- [5] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5g of terrestrial mobile telecommunication," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, 2014.
- [6] Small Cell Forum, "Small cell forum to play key role in mobile edge computing," <http://www.smallcellforum.org/blog/small-cell-forum-play-key-role-mobile-edge-computing/>, 2015.
- [7] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [8] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
- [9] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
- [10] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [11] C. Shen, J. Xu, and M. van der Schaar, "Silence is gold: Strategic interference mitigation using tokens in heterogeneous small cell networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1097–1111, 2015.
- [12] J. Torregoza, R. Enkhbat, and W.-J. Hwang, "Joint power control, base station assignment, and channel assignment in cognitive femtocell networks," *EURASIP Journal on wireless communications and networking*, vol. 2010, no. 1, p. 1, 2010.
- [13] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, 2013.
- [14] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, "Energy-efficient context-aware user association for outdoor small cell heterogeneous networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1614–1619.
- [15] C. Bottai, C. Cicconetti, A. Morelli, M. Rosellini, and C. Vitale, "Energy-efficient user association in extremely dense small cell networks," in *Networks and Communications (EuCNC), 2014 European Conference on*. IEEE, 2014, pp. 1–5.
- [16] S. Althunibat, K. Kontovasilis, and F. Granelli, "A handover policy for energy efficient network connectivity through proportionally fair access," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*. VDE, 2014, pp. 1–6.
- [17] C. Shen, C. Tekin, and M. van der Schaar, "A non-stochastic learning approach to energy efficient mobility management," *IEEE Journal on Selected Areas in Communications*, 2016.
- [18] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.
- [19] S. Ren and Y. He, "Coca: Online distributed resource management for cost minimization and carbon neutrality in data centers," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 39.
- [20] E. U. T. R. Access, "Further advancements for e-utra physical layer aspects," *3GPP Technical Specification TR*, vol. 36, p. V2, 2010.