Computation Peer Offloading in Mobile Edge Computing with Energy Budgets

Lixing Chen*, Jie Xu*, Sheng Zhou†,
*Department of Electrical and Computer Engineering, University of Miami, USA
†Department of Electronic Engineering, Tsinghua University, China

***Abstract*—The dense deployment of small-cell base stations (SBSs) endowed with cloud-like computing capabilities paves the way for pervasive mobile edge computing (MEC), enabling ultra-low latency and location-awareness for emerging mobile applications. To handle spatially imbalanced computation workloads in the network, cooperation among SBSs via peer offloading is essential to avoid large latency at overloaded SBSs and provide high quality of service to end users. However, performing effective peer offloading faces many challenges due to uncertainties of the system dynamics, limited energy budget committed by SBS owners and co-provisioning of radio access and computing services. This paper develops a novel online SBS peer offloading framework, called OPEN, by leveraging the Lyapunov technique, in order to maximize the long-term system performance while keeping the energy consumption of SBSs below individual long-term energy budget. OPEN works online without requiring future information of system dynamics, yet provides provably near-optimal performance compared to the oracle solution with complete future information. Extensive simulations are carried out and show that proposed algorithm dramatically improves the performance of edge computing system.**

I. Introduction

Pervasive mobile devices and the Internet of Things are driving the development of many resource-demanding applications. Although cloud computing enables convenient access to a centralized pool of configurable and powerful computing resources, it often cannot meet the stringent requirements of latency-sensitive applications due to the often unpredictable network latency and expensive bandwidth [1], [2]. As a remedy to these limitations, mobile edge computing (MEC) [2] has recently emerged as a new computing paradigm to enable in-situ data processing at the network edge, in close proximity to mobile devices. Considered as a key enabler of MEC, small-cell base stations (SBSs), such as femtocells and picocells, endowed with cloud-like computing and storage capabilities can serve end users' computation requests as a substitute of the cloud [3]. Nonetheless, compared to mega-scale data centers, SBSs are limited in their computing resources. Since the computation workload arrivals in small cell networks can be highly dynamic and heterogeneous, it is very difficult for an individual SBS to provide satisfactory computation service at all times. In order to overcome these difficulties, cooperation among SBSs can be exploited to enhance MEC performance and improve the efficiency of system resource utilization via computation workload peer offloading (see Figure 1 for an illustration). Although there have been quite a few works on geographical load balancing in data centers, performing workload peer offloading for MEC-enabled small cell networks faces unique challenges as follows.

This work of S. Zhou is sponsored in part by the Nature Science Foundation of China (No. 61571265, No. 91638204, No. 61621091).

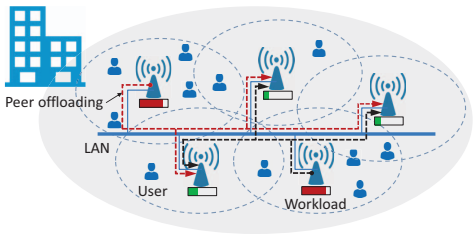


Fig. 1. Illustration of SBS peer offloading

First, small cells are often owned and deployed by individual users. Although incentive mechanism design plays an important role in motivating self-interested users to participate in the collaboration, an equally important problem is how to maximize the value of the limited resources devoted by individual SBS owners. Second, small cells operate in a highly stochastic environment with random workload arrivals. Therefore, the long-term system performance is more relevant than the immediate short-term performance. However, the limited energy budgets devoted by the SBS owners make the peer offloading decisions across time intricately intertwined, yet the decisions have to be made without foreseeing the far future. Third, whereas data centers manage only the computing resources, moving the computing resources to the network edge leads to the co-provisioning of radio access and computing services, thus mandating a new model for understanding the interdependency between two kinds of resources. In this paper, we aim to address the aforementioned challenges and our main contributions are summarized as follows:

(1) We develop a novel framework, called OPEN (Online PEer OffloadiNg), for online computation peer offloading among a network of MEC-enabled SBSs by leveraging the Lyapunov optimization [4]. We prove that the proposed algorithm achieves within a bounded deviation from the optimal system performance while restricting the potential violation of the energy budgets.

(2) We theoretically characterize the optimal peer offloading strategy. We show that the role that a SBS plays in peer offloading is determined by its pre-offloading marginal compu-

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

tation cost – a critical quantity that captures both the computation delay and energy consumption of SBSs. The optimal peer offloading strategy is to make the post-offloading marginal computation costs of the SBSs more evenly distributed.

(3) We perform extensive simulations to evaluate the performance of OPEN and verify our analytical results for various system configurations. The results confirm that our method significantly improves the system performance in terms of latency reduction and energy efficiency.

The rest of this paper is organized as follows. Section II reviews related works. Section III presents the system model and formulates the problem. Section IV develops the OPEN framework and presents the centralized solution for workload peer offloading. Simulations are carried out in Section V, followed by the conclusion in Section VI.

II. Related Work

To cope with the limited resources at MEC-enabled SBSs [5], [6], many recent works investigate SBS cooperation for improving the system performance, subject to various constraints considering radio resources [7] and energy consumption [8]). However, they focus on optimizing the radio access only without considering the computing capability of SBSs.

Computation load distribution among the network of SBSs is investigated in [9] by considering both radio and computational resource constraints. Clustering algorithms are proposed to maximize users' satisfaction ratio while keeping the communication power consumption low. In [10], coalition game is used to study computation resource sharing among femtocells. Distributed femto-cloud formation algorithm is proposed to maximize the utility of femto-clouds with constraints on fair division of incentives among femtocells. However, these works perform myopic optimization without considering the stochastic nature of the system whereas our paper studies a problem that is highly coupled across time because of individual SBS's long-term energy budgets.

Our work is closely related to geographical load balancing techniques originally proposed for data centers to deal with spatial diversities of workload patterns [11] and electricity prices [12]. Most of these works study load balancing problems that are independent across time [13]. Very few works consider temporally coupled problems. In [11], the temporal dependency is due to the switching costs of data center servers, which significantly differs from our considered problem. The closest work to our paper is [14], which aims to minimize the long-term operational cost of data centers subject to a long-term water consumption constraint. However, the constraint is imposed on the entire system whereas in our paper, each SBS has an individual energy budget.

III. System Model

We consider N SBSs (e.g. femtocells) deployed in a building (residential or enterprise) which are connected by the same Local Area Network (LAN). These SBSs are endowed with, albeit limited, edge computing capabilities and hence, end users can offload computation tasks to SBSs via wireless links. The edge computing capabilities of SBS i is characterized by computation service rate μ_i. The computation service rates of all SBSs are collected as $\boldsymbol{\mu} = \{\mu_1, ..., \mu_N\}$. To facilitate analysis, we divide the entire served area into M disjoint regions, indexed by the set $\mathcal{M} = \{1, 2, ..., M\}$. Instead of focusing on individual users, we consider computation workload coming from these regions. Due to the dense deployment of SBSs, each region $m \in \mathcal{M}$ can be covered by a subset of SBSs, denoted by $\mathcal{S}_m \subseteq \mathcal{N}$.

A. Workload arrival model

The operational timeline is discretized into time slots. In each time slot t, computation workload originating in region m is generated according to a Poisson process with arrival rate $\pi_m^t \in [0, \pi_{\max}]$. We assume that the size (in terms of Mbit) of each workload is a unit size in expectation. Let $\boldsymbol{\pi}^t = (\pi_1^t, ..., \pi_M^t)$ denote the overall workload arrival pattern across all regions which may vary across time slots. For simplicity, we assume that workload originating in region m are equally dispatched to serving SBSs in $\mathcal{S}_m$. Therefore, the workload arrival rate to SBS i, denote by ϕ_i^t, is $\phi_i^t = \sum_{m=1}^{M} \mathbf{1}\{i \in \mathcal{S}_m\} \frac{\pi_m^t}{|\mathcal{S}_m^t|}$. Nevertheless, our analysis will also apply to systems where users are served by the SBS with the strongest reception signal strength. We collect the computation workload arrival rates to all SBSs in $\boldsymbol{\phi}^t = [\phi_1^t, \phi_2^t, \ldots, \phi_N^t]$.

B. Transmission model

1) Wireless transmission energy consumption: Transmissions occur on both the wireless link between UEs and SBSs, and the wired link among SBSs. Usually the energy consumption of wireless transmission dominates and hence we consider only the wireless part. Given transmission power $P_{tx,i}$ of SBS i, the transmission rate is given by Shannon channel capacity: $r_{i,m}^t = W \log_2 \left(1 + \frac{P_{tx,i} H_{i,m}^t}{\sigma^2}\right)$, where W is the channel bandwidth, $H_{i,m}^t$ is the average channel condition between SBS i and region m, and σ^2 is the noise power. We consider the noise-limited setting by further assuming that SBSs operate on orthogonal channels. Suppose each transmission must meet a minimum rate requirement r_0, then the transmission power must satisfy $P_{i,m} = (2^{\frac{r_0}{W}} - 1)\sigma^2 (H_{i,m}^t)^{-1}$. The wireless transmission energy consumption $E_{tx,i}^t$ of SBS i in time slot t is due to sending downlink traffic to UEs, which contains the computation results and communication traffic. We model the downlink traffic of SBS i to region m as $\varpi_{i,m} = \gamma \cdot \mathbf{1}\{i \in \mathcal{S}_m\} \frac{\pi_m^t}{|\mathcal{S}_m^t|}$, where γ represents the relationship between the downlink traffic and the workload arrival. Note that the downlink traffic model can be altered to capture other traffic patterns without loss of generality. Then the energy consumption of SBS i for wireless transmission is $E_{tx,i}^t = \sum_{m \in \mathcal{M}} \frac{P_{i,m} \varpi_{i,m}}{r_0}$.

2) User-to-SBS transmission delay: The wireless transmission delay incurred by User-to-SBS offloading can be modeled as a M/G/1 queuing system [15], and the expected transmission delay for one Mbit workload is $D_{u,i}^t(\phi_i^t) =$

$\frac{1}{r_0}\left(1+\frac{\rho_i^t}{2(1-\rho_i^t)}\right)$. In this equation, $\rho_i^t = \mathbb{E}\{\phi_i^t\}/r_0$ is the workload arrival intensity of SBS i where the expectation is taken with respect to the workload arrival process in time slot t.

C. Computation model

1) Computation delay: The computation delay is due to the limited computing capability of SBSs. By modeling each SBS as M/M/1/ queuing system [16], we have the expected computation delay $D_{f,i}^t$ at SBS i in time slot t: $D_{f,i}^t(\omega_i^t) = \frac{1}{\mu_i - \omega_i^t}$, where ω_i^t is the amount of workloads processed at SBS i. Notice that ω_i^t may not equal ϕ_i^t since it is a result of the peer offloading, which will be elaborated shortly.

2) Computation energy consumption: The computation energy consumption at SBS i is load-depend, denoted as $E_{c,i}^t$. We consider a linear computation energy consumption function $E_{c,i}^t(\omega_i^t) = \kappa \cdot \omega_i^t$, where $\kappa > 0$ is the coefficient denoting the energy consumed for processing one Mbit workload.

D. SBS peer offloading

Considering imbalanced workload arrivals among the SBSs, computation offloading between peer SBSs can be performed to exploit underused computation resource to improve the overall system efficiency. We assume that workload can be offloaded only once: if workloads are offloaded to SBS j from SBS i, then it will be processed at SBS j and will not be offloaded further or back to SBS i. Let $\boldsymbol{\beta}_{i\cdot}^t = [\beta_{i1}^t, \beta_{i2}^t, \ldots, \beta_{iN}^t]$ denote the offloading decision of SBS i in time slot t, where β_{ij}^t denotes the fraction of workload offloaded from SBS i to SBS j (β_{ii}^t is the workload that SBS i retains). A peer offloading strategy for the whole system is therefore $\boldsymbol{\beta}^t = [\boldsymbol{\beta}_{1\cdot}^t, \ldots, \boldsymbol{\beta}_{N\cdot}^t]$. We further define $\boldsymbol{\beta}_{\cdot i}^t = [\beta_{1i}^t, \ldots, \beta_{Ni}^t]$ as the inbound workload of SBS i, namely workload offloaded to SBS i from other SBSs. Clearly, the total workload processed by SBS i is $\sum_{j=1}^N \beta_{ji}^t \triangleq \omega_i^t$. To better differentiate the two types of workload ϕ_i^t and ω_i^t, we call ϕ_i^t the *pre-offloading* workload and ω_i^t the *post-offloading* workload.

A peer offloading strategy $\boldsymbol{\beta}^t$ is feasible if it satisfies: (1) *Positivity*: $\beta_{ij}^t \geq 0, \forall i,j \in \mathcal{N}$. The offloaded workload must be non-negative. (2) *Conservation*: $\sum_{j=1}^N \beta_{ij}^t = \phi_i^t, \forall i \in \mathcal{N}$. The total offloaded workload (including the retained workload) by each SBS must equal its *pre-offloading* workload. (3) *Stability*: $\omega_i^t \leq \mu_i, \forall i \in \mathcal{N}$. The *post-offloading* workload of each SBS must not exceed its computation service rate. Let $\boldsymbol{\mathcal{B}}^t$ denote the set of all feasible peer offloading strategies.

Since the bandwidth of the LAN is limited, peer offloading also causes additional delay due to network congestion. We assume that the expected congestion delay $D_g^t(\lambda^t)$ depends on the total traffic through the LAN, denoted by $\lambda^t = \sum_{i=1}^N \lambda_i^t$, where $\lambda_i^t = \phi_i^t - \beta_{ii}^t$ is the amount of workload offloaded to other SBSs by SBS i. The congestion delay is modeled as a M/M/1 queuing system [16] as follows:

$$D_g^t(\lambda^t) = \frac{\tau}{1-\tau\lambda^t}, \quad \lambda^t < \frac{1}{\tau} \tag{1}$$

where τ is the average communication time for sending and receiving a unit workload over the LAN without congestion.

E. Problem formulation

Peer offloading relies on SBSs' cooperative behavior in sharing their computing resources as well as their energy costs. A large body of literature was dedicated to design incentive mechanisms [17] to encourage cooperation among self-interested SBSs to improve the social welfare. The focus of our paper is not to design yet another incentive mechanism. Instead, we design SBS peer offloading strategies taking the SBS committed resources as the input and hence, our method can work in conjunction with any existing incentive mechanisms. In this paper, we assume that each SBS has a long-term energy budget as a result of some incentive mechanism. Given the system model, the total delay cost and energy consumption of SBS i in time slot t are as follows:

$$D_i^t(\boldsymbol{\beta}^t, \phi_i^t) = \sum_{j\in\mathcal{N}} \beta_{ij}^t D_{f,j}^t + \lambda_i D_g^t + \phi_i^t D_{u,i}^t \tag{2}$$

$$E_i^t(\boldsymbol{\beta}^t, \phi_i^t) = E_{tx,i}^t + E_{c,i}^t(\boldsymbol{\beta}^t) = E_{tx,i}^t + \kappa \sum_{j\in\mathcal{N}} \beta_{ji}^t \tag{3}$$

The objective of the network operator is to minimize the long-term system delay given the energy consumption budgets committed by individual SBSs (which are outcomes of the adopted incentive mechanisms). Formally, the problem is

$$\boldsymbol{\mathcal{P}1} \quad \lim_{T\to\infty} \min_{\boldsymbol{\beta}^t \in \boldsymbol{\mathcal{B}}^t} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\left\{D_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t)\right\}$$

$$\text{s.t.} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{E_i^t(\boldsymbol{\beta}^t, \phi_i^t)\right\} \leq \bar{E}_i, \forall i \in \mathcal{N} \tag{4}$$

$$E_{tx,i}^t(\boldsymbol{\phi}_i^t) + E_{c,i}^t(\boldsymbol{\beta}^t) \leq E_{\max}, \forall i \in \mathcal{N}, \forall t \tag{5}$$

$$D_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t) \leq D_{\max}, \forall i \in \mathcal{N}, \forall t \tag{6}$$

Constraint (4) is the long-term energy budget for each SBS. Constraint (5) requires that the energy consumption of a SBS does not exceed an upper limit $E_{\max}$ in each time slot. Constraint (6) indicates that the per-slot delay of each SBS is capped by an upper limit $D_{\max}$ so that the real-time performance is guaranteed in the worst case.

Otimally solving $\boldsymbol{\mathcal{P}1}$ requires complete offline information (distribution of workload arrivals in all time slots) which is difficult to predict in advance, if not impossible. Moreover, the long-term energy budget couples the peer offloading decision across different slots: consuming more energy at current time will potentially reduce the available energy for future use. These challenges call for an online approach without foreseeing the future.

IV. Online SBS Peer Offloading

In this section, we develop OPEN (Online SBS PEer offloadiNg) by leveraging the Lyapunov optimization. OPEN converts $\boldsymbol{\mathcal{P}1}$ to per-slot optimization problems and offers a feasible solution requiring only current information.

A. Lyapunov optimization based online algorithm

We leverage the Lyapunov optimization technique [4] and construct a (virtual) energy deficit queue for each SBS to guide the SBS peer offloading decisions to follow the long-term energy budget. Let $\boldsymbol{q}(t) = [q_1(t), q_2(t), \ldots, q_N(t)]$ denote a set of energy deficit queues, one for each SBS, and assume $q_i(0) = 0, \forall i \in \mathcal{N}$. The energy deficit queue of SBS i evolves:

$$q_i(t+1) = \max\{q_i(t) + E_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t) - \bar{E}_i, 0\} \tag{7}$$

where $q_i(t)$ is the queue length indicating the deviation of current energy consumption from the energy budget of SBS i.

Next, we present the online algorithm OPEN (in Algorithm 1) for solving $\boldsymbol{\mathcal{P}1}$. In OPEN, the network operator determines the peer offloading strategy in each time slot t by solving the optimization problem $\boldsymbol{\mathcal{P}2}$ presented below:

$$\boldsymbol{\mathcal{P}2} \quad \min_{\boldsymbol{\beta}^t \in \boldsymbol{\mathcal{B}}^t} \sum_{i=1}^{N} \left(V \cdot D_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t) + q_i(t) \cdot E_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t)\right)$$
$$\text{s.t.} \qquad (5) \text{ and } (6)$$

The positive control parameter V is used to adjust the trade-off between system delay cost and the energy consumption of SBSs. Notice that solving $\boldsymbol{\mathcal{P}2}$ requires only currently available information. By considering the additional term $\sum_{i=1}^{N} q_i(t) E_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t)$ in $\boldsymbol{\mathcal{P}2}$, the network operator takes into account the energy deficits of SBSs during current-slot peer offloading. When $\boldsymbol{q}(t)$ is larger, minimizing the energy deficits is more critical, thereby guiding the SBSs towards meeting the energy budge and enabling online decision. In Theorem 1, we give a firm performance bound of OPEN.

Algorithm 1: OPEN

Input: control parameter V, virtual queues $\boldsymbol{q}(0) = \mathbf{0}$;
Output: offloading decisions $\boldsymbol{\beta}^0, \ldots, \boldsymbol{\beta}^{T-1}$;
for $t = 0$ ***to*** $T-1$ **do**
  Observe workload arrival $\boldsymbol{\phi}^t$ and feasible peer offloading strategy set $\boldsymbol{\mathcal{B}}^t$;
  Solving $\boldsymbol{\mathcal{P}2}$ to get optimal $\boldsymbol{\beta}^t$ in time slot t;
  $q_i(t+1) = [q_i(t) + E_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t) - \bar{E}_i]^+$
end
return $\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^{T-1}$;

Theorem 1. *By applying OPEN, the time-average system delay satisfies:*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\left\{D_i^t(\boldsymbol{\beta}^t, \boldsymbol{\phi}^t)\right\} < D_{sys}^{opt} + \frac{B}{V}$$

and the time-average energy consumption of SBSs satisfies:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\left\{E_i(t)\right\} \leq \frac{B + V(D_{sys}^{max} - D_{sys}^{opt})}{\epsilon} + \bar{E}_{sys}$$

where $B = \frac{1}{2}\sum_{i=1}^{N}(E_{\max} - \bar{E}_i)^2$; D_{sys}^{opt} is the optimal system delay $\boldsymbol{\mathcal{P}1}$; $D_{sys}^{max} = N D_{max}$ is the largest system delay; $\bar{E}_{sys} = \sum_{i=1}^{N} \bar{E}_i$ is system energy budget; $\epsilon > 0$ is a constant.

Proof. See online Appendix [18]. □

Theorem 1 demonstrates an $[O(1/V), O(V)]$ delay-energy tradeoff. OPEN asymptotically achieves the optimal performance of the offline problem $\boldsymbol{\mathcal{P}1}$ by letting $V \to \infty$. However, optimal performance is achieved at the price of a higher energy consumption, as a large energy deficit queue is required to stabilize the system and hence postpones the convergence.

B. Solving the per time slot problem

To complete OPEN, it remains to solve the optimization problem $\boldsymbol{\mathcal{P}2}$. We consider the existence of a centralized controller who collects complete current-slot information from all SBSs, solves $\boldsymbol{\mathcal{P}2}$ and coordinates the SBS peer offloading in each time slot. For the ease of exposition, we drop the time index t in this subsection. Before proceeding, we simplify $\boldsymbol{\mathcal{P}2}$ by considering only the decision-dependent parts:

$$\min_{\boldsymbol{\beta}^t \in \boldsymbol{\mathcal{B}}^t} \sum_{i\in\mathcal{N}} \left(\frac{V\omega_i}{\mu_i - \omega_i} + \kappa q_i \omega_i\right) + \frac{V\tau\lambda}{1-\tau\lambda}$$

Recall $\omega_i = \sum_{j=1}^{N} \beta_{ji}$. Although the decision-independent parts (i.e. UE-to-SBS transmission delay and energy consumption) do not affect the solution of $\boldsymbol{\mathcal{P}2}$ directly, they will affect the energy deficit queue updating and hence indirectly affects the SBS peer offloading decisions in the long-run.

Next, we present the optimal solution for the above optimization problem starting with classifying SBSs into the following three categories similar to [19]: (1) **Source SBS** ($\mathcal{R}$): offloads a portion of its *pre-offloading* workloads to other SBSs and processes the rest of workloads locally. It does not receive any workload from other SBSs. ($0 \leq \omega_i < \phi_i$); (2) **Neutral SBS** ($\mathcal{U}$): processes all its *pre-offloading* workloads locally and does not receive any workload from other SBSs. ($\omega_i = \phi_i$); (3) **Sink SBS** ($\mathcal{S}$): receives workloads from other SBSs and does not offload workload to others. ($\omega_i > \phi_i$). Notice that there is no SBS such that it offloads workloads to other SBSs while receiving workloads from other SBSs. It can be easily shown that having such SBSs result in suboptimal solutions due to the extra network congestion cost.

To assist the presentation of the optimal solution, we define two auxiliary functions: define $d_i(x_i) \triangleq \left.\frac{\partial[\omega_i D_{f,i}(\omega_i)]}{\partial \omega_i}\right|_{\omega_i = x_i} = \frac{\mu_i}{(\mu_i - x_i)^2}$ as the *Marginal Computation Delay function*; $g(\lambda) \triangleq \frac{\partial}{\partial\lambda}[\lambda D_g(\lambda)] = \frac{\tau}{(1-\tau\lambda)^2}$ as the *Marginal Congestion Delay function*. Let $\xi_i \triangleq V d_i(\phi_i) + \kappa q_i$ be the *pre-offloading* marginal computation cost (MCC), taking into account both the computation delay cost and the computation energy cost if SBS i processes all its workload locally. Theorem 2 shows how the categorization of SBSs and workload allocation can be decided based on ξ_i.

Theorem 2. *The category that SBS i ($i \in \mathcal{N}$) belongs to and the post-offloading workload ω_i can be decided as follows:*
(a) If $\xi_i < \alpha$, then $i \in \mathcal{S}$ and $\omega_i = d_i^{-1}(\frac{1}{V}(\alpha - \kappa q_i))$
(b) If $\alpha \leq \xi_i \leq \alpha + V g(\lambda)$, then $i \in \mathcal{U}$ and $\omega_i = \phi_i$
(c) If $\xi_i > \alpha + V g(\lambda)$, then $i \in \mathcal{R}$ and

$\omega_i = [d_i^{-1}(\frac{1}{V}(\alpha + Vg(\lambda) - \kappa q_i))]^+$;
where λ, α *are the solution to the workload flow equation*

$$\underbrace{\sum_{i\in\mathcal{S}}\left(d_i^{-1}(\frac{1}{V}(\alpha-\kappa q_i))-\phi_i\right)}_{\lambda^S\text{: inbound workloads to sinks}} = \underbrace{\sum_{i\in\mathcal{R}}\left(\phi_i-[d_i^{-1}(\frac{1}{V}(\alpha+Vg(\lambda)-\kappa q_i))]^+\right)}_{\lambda^R\text{: outbound workloads from sources}} \quad (8)$$

Proof. See online Appendix [18]. □

In Theorem 2, α corresponds to the unique optimal *post-offloading* MCC (i.e. $Vd_i(\omega_i) + \kappa q_i$) of sink SBSs. Part ($a$) indicates that SBSs with *pre-offloading* MCC less than α will serve as sink SBSs and their post-offloading MCCs will be equal to α. Part (b) implies that the *pre-offloading* MCC of neutral SBS is no less than α but no larger than the sum of α and the marginal congestion delay cost. This means that no other SBSs will benefit from offloading workloads to neutral SBSs and at the same time neutral SBSs receive no benefits by performing peer offloading. For source SBS, its *pre-offloading* MCC is larger than the sum of α and the marginal congestion delay cost and therefore, it tends to offload workloads to other SBSs until its *post-offloading* MCC reduces to $\alpha + Vg(\lambda)$.

We develop a simple iterative algorithm to obtain α using binary searching under workload flow equation (summarized in Algorithm 2). In each iteration, the algorithm first determines a set of sink SBSs ($\mathcal{S}$) according to the parameter α, then the corresponding amount of inbound workload is determined. Given the total workloads $\lambda = \lambda^S$ transmitted in the LAN, the algorithm determines the source SBSs $\mathcal{R}$, neutral SBSs $\mathcal{U}$ and then calculates the outbound workload λ^R. If λ^R equals λ^S, then the optimal α is found; otherwise, the algorithm updates α and goes into the next iteration.

Algorithm 2: Algorithm for Solving $\mathcal{P}2$

Input: $\boldsymbol{\mu}^t$; $\boldsymbol{\phi}^t$; τ.
Output: Load allocation to SBSs: $\omega_1, \omega_2, \ldots, \omega_N$.
Initialization: $\omega_i \leftarrow \phi_i, i \in \mathcal{N}$;
Sort SBSs: $Vd_1(\phi_1) + \kappa q_1 \leq \ldots \leq Vd_N(\phi_N) + \kappa q_N$;
If $Vd_1 + Vg(0) + \kappa q_1 \geq Vd_N(\phi_N) + \kappa q_N$ **STOP** (no peer offloading is required);
$a \leftarrow Vd_1(\phi_1) + \kappa q_1$; $b \leftarrow Vd_N(\phi_N) + \kappa q_N$;
while $|\lambda^S(\alpha) - \lambda^R(\alpha)| \geq \epsilon$ **do**
  $\lambda^S(\alpha) \leftarrow 0$, $\lambda^R(\alpha) \leftarrow 0$;
  $\alpha \leftarrow \frac{1}{2}(a+b)$;
  Calculate: $\mathcal{S}(\alpha)$, $\lambda^S(\alpha)$, $\mathcal{R}(\alpha)$, $\mathcal{U}(\alpha)$, $\lambda^R(\alpha)$;
  if $\lambda^S(\alpha) > \lambda^R(\alpha)$ **then** $b \leftarrow \alpha$;
  else $a \leftarrow \alpha$;
end
Determine the workload allocation ω_i;

V. Simulation

In this section, we evaluate the performance of the proposed OPEN algorithm through simulations under various system settings. We consider SBSs deployed in a building which are connected by the same LAN. Our simulation adopts the stochastic geometry approach for SBS deployment, which is modeled as a homogeneous Poisson Point Process (PPP) [20]. The PPP model captures the fact that SBSs are randomly deployed in heterogeneous networks. Specifically, we simulate a 100m×100m area served by a set of SBSs whose locations are chosen according to the PPP process with density $\lambda_p = 0.4$. The whole area is regularly divided into 25 square regions. Each region is served by SBSs in the vicinity. End users offload their computation tasks to SBSs within communication distance $d_{com} = 20$m. Workloads arrive at each region following a Poisson process with arrival rate $\pi_m^t \in [0, 20]$ Mbit/sec. The service rate of each SBS is μ_i =20 Mbit/sec. The mean communication time within the LAN is set as $\tau = 0.01$ sec. Other default parameters for the simulation are: minimum transmit rate $r_0 = 30$Mbps, channel bandwidth $W = 20$MHz, downlink traffic coefficient $\gamma = 1$.

The performance of proposed methods is compared with two benchmark schemes: (1) **No peer offloading**: no SBS peer offloading is performed in the network, i.e. SBSs process all workload received from the end users locally. (2) **Delay-Optimal**: We use the load balancing method in [19]. In each slot we regard the SBS peer offloading as a static load balancing problem to achieve the lowest system delay without considering the energy budgets of SBSs.

A. Run-time Performance Evaluation

Figure 2 shows the system delay cost of OPEN and benchmark schemes in the first 50 time slots. As can be seen, the network suffers from large system delay without SBS peer offloading. With peer offloading technique, delay-optimal achieves the best delay performance since it ignores the energy budget and OPEN achieves slightly worse system delay performance in order to satisfy the energy budgets of SBSs. Figure 3 and 4 show the time average system delay and time average energy budget violation. It is shown that without peer offloading the network bears both high system delay and energy violation due to the uneven workload allocation. Even though delay-optimal successfully reduces the system delay using peer offloading, it incurs considerable energy budget violation. By contrast, OPEN achieves low system delay similar to the delay-optimal strategy while satisfying the energy budget with small violation.

B. Effect of System Utilization

System utilization χ is defined as the ratio of the expected total workload arrival rate to the aggregate service rate of the SBS network: $\chi = \mathbb{E}\{\sum_{i=1}^N \phi_i\}/(\sum_{i=1}^N \mu_i)$. Figure 5(a) presents the system delay for different values of system utilization ranging from 10% to 99%. It shows that peer offloading strategies help reduce the system delay for all levels of system utilization. Moreover, the system benefits more from

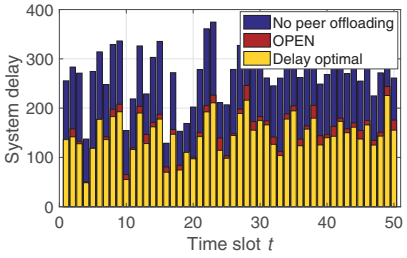


Fig. 2. Per-slot system delay

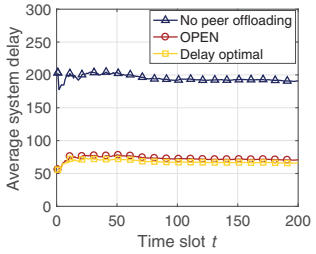


Fig. 3. Time average system delay

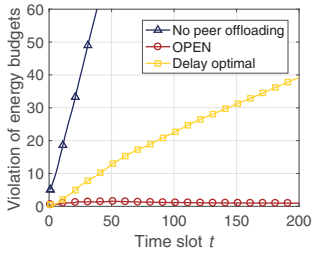
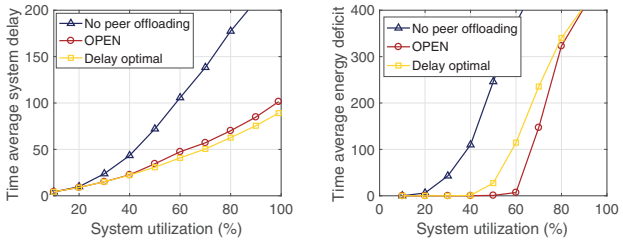


Fig. 4. Time-average energy budget violation

SBS peer offloading in the high system utilization regime. This is due to the fact that the number of overloaded SBS increases as the system utilization goes high. It also shows that when the system utilization is less than 50%, our method achieves almost identical performances compared to the delay-optimal scheme since low workload arrival leads to small energy deficit queue $\boldsymbol{q}$. As $\boldsymbol{q}$ goes to 0, the proposed schemes have the identical optimization objectives as delay-optimal scheme.

Figure 5(b) shows the system energy deficits under different system utilization. The figure illustrates the capability of four schemes in terms of holding the energy budgets. The energy budgets are set to be the same for all system utilization levels. Without peer offloading, the energy budgets are violated at a low system utilization ratio (20%). Delay-optimal strategy helps to hold the energy budgets for SBSs by balancing the workload and increases the violation point to 40% system utilization. OPEN further increase the violation point to 60%.



(a) System delay (b) Energy deficit

Fig. 5. Effect of system utilization

VI. Conclusion

In this paper, we investigated peer offloading schemes in MEC-enabled small cell networks where heterogeneous workload arrival pattern in both spatial and temporal domains is envisioned. We developed OPEN, a novel online peer offloading framework, to optimize the edge computing performance under limited energy budgets committed by individual SBSs without forseeing future information. The proposed framework provides provable performance guarantee compared to the oracle solution that has the complete future information. Our model and analysis not only provide important insights for designing efficient MEC-enabled small cell networks but also can be adapted to improve the performance of many other MEC systems involving cooperation among multiple entities.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[2] 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, 2016.

[3] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, 2016.

[4] 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.

[5] TROPIC. (2015) Distributed computing storage and radio resource allocation over cooperative femtocells. [Online]. Available: http://www.ict-_tropic.eu.

[6] F. Lobillo, Z. Becvar, M. A. Puente, P. Mach, F. L. Presti, F. Gambetti, M. Goldhamer, J. Vidal, A. K. Widiawan, and E. Calvanesse, "An architecture for mobile computation offloading on cloud-enabled lte small cells," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE*. IEEE, 2014, pp. 1–6.

[7] A. Abdelnasser, E. Hossain, and D. I. Kim, "Clustering and resource allocation for dense femtocells in a two-tier cellular ofdma network," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1628–1641, March 2014.

[8] J. Rubio, A. Pascual-Iserte, J. del Olmo, and J. Vidal, "User association for load balancing in heterogeneous networks powered with energy harvesting sources," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 1248–1253.

[9] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–6.

[10] S. Tanzil, O. Gharehshiran, and V. Krishnamurthy, "A distributed coalition game approach to femto-cloud formation," *IEEE Transactions on Cloud Computing*, 2016.

[11] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing," in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10.

[12] J. Luo, L. Rao, and X. Liu, "Spatio-temporal load balancing for energy cost optimization in distributed internet data centers," *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 387–397, July 2015.

[13] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 233–244.

[14] M. Islam, S. Ren, G. Quan, M. Shakir, and A. Vasilakos, "Water-constrained geographic load balancing in data centers," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[15] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[16] R. B. Cooper, *Introduction to queueing theory*. North Holland, 1981.

[17] Y. Zhang and M. v. d. Schaar, "Incentive provision and job allocation in social cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 607–617, September 2013.

[18] L. Chen. (2017) Online appendix for this paper. [Online]. Available: https://www.dropbox.com/s/kjxfykgsxtvqh1i/AppendixOPEN.pdf?dl=0

[19] S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 4, pp. 537–555, 2011.

[20] F. Baccelli, B. Błaszczyszyn *et al.*, "Stochastic geometry and wireless networks: Volume ii applications," *Foundations and Trends® in Networking*, vol. 4, no. 1–2, pp. 1–312, 2010.