Lifetime Maximization in Mobile Edge Computing Networks

Sabyasachi Gupta and Jacob Chakareski, *Senior Member, IEEE*

***Abstract*—Mobile edge computing has emerged as a promising technology to augment the computational capabilities of mobile devices. For a multi-user network in which its users periodically compute their tasks with the help of an edge cloud, we investigate the network lifetime maximization problem based on present user task information. We pursue this objective via a minimum energy efficiency maximization (MEEM) strategy that jointly optimizes the fraction of user task computations offloaded to the cloud and the respective allocation of edge computing and network communication resources across the users. We also investigate the network lifetime maximization problem for the case when the user task information is available for all future time slots, as well. This setting represents an upper bound for the MEEM strategy. Optimal solutions for both investigated strategies are formulated via feasibility testing and geometric programming. We show that MEEM can achieve a 70% lifetime improvement over the state-of-the-art and 460% lifetime improvement over the case of local user task computation only. We also show that for a high value of the maximum tolerable delay for completing the computation tasks of the users, MEEM achieves the globally optimal network lifetime performance. Finally, we show that MEEM achieves a significant reduction (3X) in variation of enabled network lifetime over diverse network topologies, relative to the state-of-the-art.**

***Index Terms*—Mobile-edge computing, energy efficiency, lifetime maximization, resource allocation.**

I. Introduction

As mobile devices are gaining enormous popularity over the last decade, many new applications, e.g., virtual reality, natural language processing, interactive gaming, speech-to-text, image processing, have emerged and attracted great attention. Due to the requirements of high reliability, intensive computing, and low latency for these applications, the concept of Mobile-Edge Computing (MEC) has emerged [2]. In MEC based systems, small-scale cloud-computing facilities are available at the edge of pervasive radio access networks in close proximity to the mobile users [2].

Manuscript received June 12, 2019; revised September 16, 2019 and November 22, 2019; accepted December 11, 2019. Date of publication January 10, 2020; date of current version March 12, 2020. This work was supported in part by NSF Awards CCF-1528030, ECCS-1711592, CNS-1836909, and CNS-1821875 and in part by research gifts and an Adobe Data Science Award from Adobe Systems. This paper was presented in part at the IEEE Global Communications Conference, Waikoloa, HI, USA, Dec., 2019 [1]. The review of this article was coordinated by Dr. S. Misra. *(Corresponding author: Sabyasachi Gupta.)*

S. Gupta is with the Department Electrical Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: sabyasachig@smu.edu).

J. Chakareski is with the Ying Wu College of Computing, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: jacob@ua.edu).

Digital Object Identifier 10.1109/TVT.2020.2965440

A. Motivation

In this paper, we investigate joint computing task sharing and computing and communication resource allocation in mobile edge computing networks, towards maximizing their lifetime. To the best of our knowledge, lifetime maximization has not been explored for such networks before. In particular, though prior studies have examined energy efficiency in mobile-edge computing networks, they have not considered the residual battery energy information for the wireless nodes, when allocating computing and communication resources in such networks [2]–[11]. Thus, these studies may not necessarily result in good (long) network lifetimes. The motivation behind our work is based on the following observations:

- To improve the lifetime of a network with battery operated nodes, the decisions on communication and computation resource allocation for the users need to be made based on the residual battery energy of their devices. For example, a node which has low residual battery energy and a highly computation-intensive task to complete, should be allocated high communication and edge cloud computation resources, so that it can compute its task with low energy consumption.
- Solving the network lifetime maximization problem requires availability of user task information for all future time slots, as shown later on. However, task information for the users may not be available for future time slots. Therefore, it is important to design a resource allocation strategy which can operate based on user task information solely for the present time slot and the current residual battery energy information for the users.

B. Contributions

The scenario we investigate is illustrated in Fig. 1. Aiming to maximize the network lifetime, we investigate the joint optimization of sharing computation between the users and the edge cloud, and allocating communication and edge computing resources for each user. The lifetime of a network is defined as the time interval during which each of its users can compute his task within a maximum tolerable delay and none of the users is depleted of device battery energy. Our main contributions are:

- Aiming to maximize the network lifetime based on user task information for the present time slot only, we explore a minimum energy efficiency maximization (MEEM) strategy for joint optimization of the fraction of user task computations offloaded to the cloud and the respective

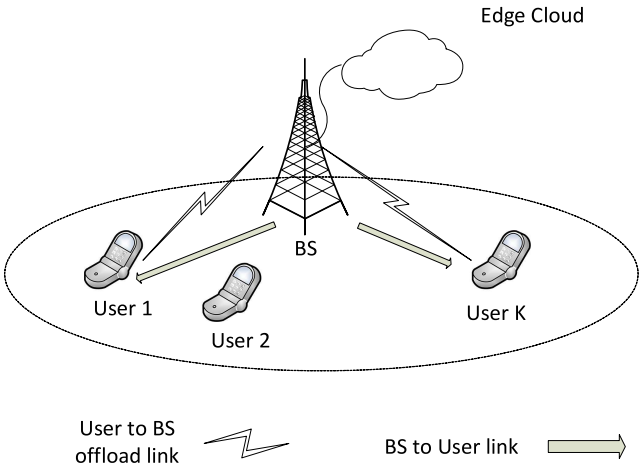

Fig. 1. System model of the scenario under investigation.

allocation of edge computing and network communication resources across the users.

- We optimally solve the network lifetime maximization problem when future user task information is available. This setting is an upper bound for MEEM. Furthermore, an upper bound to the optimal network lifetime is obtained for the case in which the task characteristic for all the users at each time slot is same.
- We formulate the optimal solutions for the proposed strategies using feasibility testing and geometric programming. We also discuss the centralized implementation of MEEM strategy.
- We show that MEEM achieves significant network lifetime improvement over local computation scheme (460%). Furthermore, we compare our proposed strategy with the following state-of-the-art methods: i) minimizing the total energy consumption of the users, and ii) minimizing the maximum energy consumption across the users and show that our proposed strategy can achieve 50–70% improvement in network lifetime compared to them.
- We show that MEEM achieves a significant reduction (3X) in variation of enabled network lifetime over diverse network topologies, relative to the state-of-the-art.

C. Related Work

Since wireless devices have limited battery energy, energy efficiency is a crucial design parameter for cooperative wireless networks. Significant effort has been made to date to investigate maximizing the lifetime of such networks [12]–[22]. Network lifetime maximization with power allocation and relay selection for the single-user cooperative network is investigated in [12]–[19]. It has been shown that the wireless node's residual battery energy information must be taken into account in deciding the transmit power control, relay selection, and channel allocation, so that the overall network lifetime is improved [19]. For multiple-user cooperative network, Himsoon *et al.* [20] has studied joint power allocation and relay placement problem for lifetime maximization. Power allocation and partner selection for lifetime maximization in pairwise cooperative network has been investigated in [22].

For wireless networks in which the nodes have computationally intensive tasks with low latency requirements, offloading them to the edge cloud may improve the network energy efficiency [2]–[11], [23]–[31]. You *et al.* [7] investigates a weighted sum energy consumption minimization scheme in mobile-edge computing networks, by jointly optimizing the load and communication resource allocation. A joint optimization of the utilization of radio resources, the transmit precoding matrices of the users, and the allocation of computational resources is proposed for MIMO multi-cell systems with the aim of minimizing the overall user energy consumption, while meeting the latency constraints for each user's task [8]. For a multi-server mobile-edge computing network, Tran *et al.* [10] studies a joint computation resource allocation, transmit power allocation, and task offloading decision optimization, to minimize a system utility casted as a weighted function of the task completion time and task energy consumption. Cao *et al.* [11] investigate computation and communication resource allocation when task is computed with help of a peer device and edge cloud to minimize the total energy consumption in the network while satisfying the users computation latency constraint.

D. Organization of the Paper

The rest of this paper is organized as follows. In Section II, we describe our system models. The joint optimization of computation task sharing and resource allocation for the proposed MEEM strategy is formulated in Section III. This section also includes a formulation of the network lifetime maximization problem with the availability of future user task information. We derive the optimal solutions via geometric programming for all three strategies under investigation in Sections IV-V, respectively. Numerical simulation results are examined in Section VI. The paper concludes in Section VII.

II. System Model

Our multiuser network comprises of K users denoted by the set $\mathcal{K} = \{1, .., K\}$ and a base station (BS) equipped with an edge cloud of limited computational capability. Each user $k \in \mathcal{K}$ has a computation capability of f_k and initial battery energy e_k J. The system operates in a time-slotted manner where in every n seconds, the edge cloud serves a set of users which have computationally intensive tasks. We consider a quasi-static scenario where the set of mobile users remains unchanged during a computation offloading period, while may change across different time-slots.

Let $\mathcal{K}_l \subseteq \mathcal{K}$ denote the set of users to be served by the cloud at slot $l \in \{1, 2, ..\}$. Let user $k \in \mathcal{K}_l$ has a task $\phi_k(l) = (\beta_k(l), b_k(l))$ to compute at the lth time slot, where $b_k(l)$ is the number of bits to be computed which include program codes, and input parameters and $\beta_k(l)$ is the required number of CPU cycles for 1 bit computation of the task. Therefore $\beta_k(l)b_k(l)$ denotes the total CPU cycles required to compute the task $\phi_k(l)$. The method proposed in [32] can be applied to determine $b_k(l)$ and $\beta_k(l)$. In [33], authors have investigated the value of $\beta_k(l)$

TABLE I
MAJOR NOTATION USED IN THE PAPER

Parameters	Definition
$\mathcal{K}$	Set of users
f_k	CPU frequency of user k
e_k	Initial battery energy of user k
n	Duration of slots in sec.
$\mathcal{K}_l$	Set of active users at time slot l
$\phi_k(l)$	Task of user k at time slot l
$\beta_k(l)$	Number of CPU cycles required for 1 bit computation of task $\phi_k(l)$
$b_k(l)$	Number of bits to be computed for the task $\phi_k(l)$
T^{th}	Maximum tolerable delay for the tasks
$T_k(l)$	Local computation time of user k at time slot l
$b_k^{\mathrm{EC}}(l)$	Number of bits offloaded to the edge cloud by user k at time slot l
$E_k(l)$	Energy consumption for local computation for user k at time slot l
γ_c	Effective switched capacitance of the CPU
$R_{k,b}$	Spectral efficiency through the link between user k and BS
P_k	Transmit power density of user k
$g_{k,b}$	Large-scale channel gain from user k to BS
N_0	Noise power spectral density
$\tau_{k,\mathrm{EC}}(l)$	Delay in offloading $b_k^{\mathrm{EC}}(l)$ bits for user k
$B_k(l)$	Bandwidth allocated to user k at time slot l
$T_{\mathrm{EC},k}(l)$	Computation time for computing $b_k^{\mathrm{EC}}(l)$ bits at the edge cloud
$F_k(l)$	computation resource allocated to user k at time slot l
$e'_k(l)$	Residual energy of the user k at time slot l

for some of the applications. Similar to [6], [7], [28]–[31], we consider splittable task and therefore each user can fully or partially offload its computing tasks to the BS. The tasks are needed to be executed within a maximum tolerable delay $\mathsf{T}^{th} \leq n$. An example of such network is internet of things (IoT) networks in which the edge cloud receives periodically splittable task, e.g., images from the IoT devices for processing. Table I summarizes the main notation used in the paper.

A. Local Computation

As shown in Fig. 2, user $k \in \mathcal{K}_l$ offloads $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud and computes $b_k(l) - b_k^{\mathrm{EC}}(l)$ bits at its own processor at time slot l. Thus, the local computation time is

$$T_k(l) = \frac{\beta_k(l)\left(b_k(l) - b_k^{\mathrm{EC}}(l)\right)}{f_k}. \tag{1}$$

Following the standard energy consumption model for task computation in [34], the overall computation energy at user k to compute $b_k(l) - b_k^{\mathrm{EC}}(l)$ bits is

$$E_k(l) = \gamma_c \beta_k(l)\left(b_k(l) - b_k^{\mathrm{EC}}(l)\right) f_k^2, \tag{2}$$

where γ_c is the effective switched capacitance of the CPU.

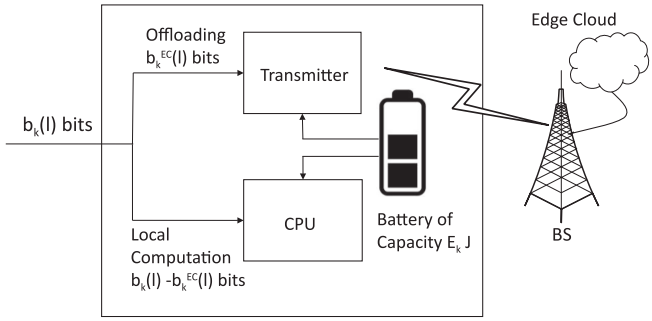

Fig. 2. Task computation of user k assisted by the edge cloud.

B. Computation of Offloaded Tasks

Each user $k \in \mathcal{K}_l$ offloads $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud at time slot l, and then the edge cloud computes these bits at its processor and sends back the output of the computed tasks to the users. Let the bandwidth allocated to user k at time slot l be $B_k(l)$. The spectral efficiency (in b/s/Hz) of the link between user k and the base station, for ergodic Rayleigh fading, is [35]:

$$R_{k,b} = \exp\left(\frac{N_0}{P_k\, g_{k,b}}\right) E_1\left(\frac{N_0}{P_k\, g_{k,b}}\right) \log_2 e \tag{3}$$

where $E_1(x) = \int_1^\infty m^{-1} e^{-xm} dm$ is an exponential integral, $g_{k,b}$ is the large-scale channel gain from user k to the BS, P_k is the transmit power density of user k, and N_0 is the noise power spectral density. Therefore, the delay in offloading $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud becomes

$$\tau_{k,\mathrm{EC}}(l) = \frac{b_k^{\mathrm{EC}}(l)}{B_k(l) R_{k,b}}, \tag{4}$$

The energy consumption at user k to offload $b_k^{\mathrm{EC}}(l)$ bits is

$$\mathcal{E}_k(l) = P_k \frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}}. \tag{5}$$

Let the cloud allocate $F_k(l)$ of its computation resource to user k at time slot l. Thus, to compute the $b_k^{\mathrm{EC}}(l)$ bits for user k, the edge cloud requires time

$$T_{\mathrm{EC},k}(l) = \frac{\beta_k(l) b_k^{\mathrm{EC}}(l)}{F_k(l)}. \tag{6}$$

III. PROBLEM FORMULATION

The overall completion time of task $\phi_k(l)$, $k \in \mathcal{K}_l$, is

$$\mathsf{T}_k(l) = \max\left(T_k(l), \tau_{k,b}(l) + T_{\mathrm{EC},k}(l)\right). \tag{7}$$

We disregard the time spent in sending back the results of the computation, as the size of the output data tends to be small relative to the input data [3].

The network lifetime is defined as the time duration for which all user tasks are executed within a maximum tolerable delay, while none of the users is depleted of energy. Thus, maximizing

the lifetime of the network can be expressed as:

$$\begin{aligned}\max_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad & \mathsf{T}, \\ \text{s.t.} \quad & \sum_{l\in S_k^{\mathsf{T}}} (E_k(l)+\mathcal{E}_k(l)) \le e_k, \quad k\in\{1,..,K\}, \\ & \mathsf{T}_i(m) \le \mathsf{T}^{th}, \quad i\in\mathcal{K}_m, \ \ m\in\{1,..,\mathsf{T}\}, \\ & \sum_{i\in\mathcal{K}_m} B_i(m) \le B, \quad m\in\{1,..,\mathsf{T}\}, \\ & \sum_{i\in\mathcal{K}_m} F_i(m) \le F, \quad m\in\{1,..,\mathsf{T}\}, \end{aligned} \tag{8}$$

where T denotes the network operating time in number of slots, S_k^{T} denotes the set of time slots when user k is activated within the network operating time T, B is the total available bandwidth in the system and F is the total processing capability of the cloud. In turn, $\boldsymbol{B}$, $\boldsymbol{F}$, and $\boldsymbol{b}$ are respectively the vectors of all values of $B_i(m)$, $F_i(m)$, and $b_i^{\text{EC}}(m)$, for $i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}\}$. The first constraint in (8) imposes that the energy consumption of user k (in local computation and offloading bits) over T be bounded by its initial battery energy e_k. The second constraint imposes that the task completion time of user i at the time slot m be bounded by the maximum tolerable delay T^{th}. The communication and computation resource allocations for the mobile users and the cloud at each time slot m are restricted by the total system bandwidth and the cloud's processing capability, respectively, as captured by the third and fourth constraints.

The above problem is hard to solve in practice for two reasons. Firstly, to obtain computation and communication resource allocation based on this strategy, task information for users in future time slots, $\beta_i(m), b_i(m), i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}\}$ needs to be available which may not be practical. Secondly, the number of optimization variables is large (proportional to T) in (8). Thus, finding the optimal solution requires very high computational complexity. Aiming to maximize the network lifetime based on user task information for the present time slot only, we investigate the following optimization problem:

$$\begin{aligned}\max_{\boldsymbol{F'},\boldsymbol{B'},\boldsymbol{b'}} \quad & \min_{k\in\mathcal{K}_l} \ \eta_k(l), \\ \text{s.t.} \quad & \mathsf{T}_k(l) \le \mathsf{T}^{th}, \ \ k\in\mathcal{K}_l, \\ \text{s.t.} \quad & \sum_{k\in\mathcal{K}_l} B_k(l) \le B, \quad \sum_{k\in\mathcal{K}_l} F_k(l) \le F, \end{aligned} \tag{9}$$

where $\eta_k(l) = e'_k(l)/(E_k(l)+\mathcal{E}_k(l))$ is the energy efficiency of user $k\in\mathcal{K}_l$ with $e'_k(l)$ as the residual energy of the user k at time slot l, and $\boldsymbol{B'}$, $\boldsymbol{F'}$, and $\boldsymbol{b'}$ are respectively the vectors of all values of $B_k(l)$, $F_k(l)$, and $b_k^{\text{EC}}(l)$, for $k\in\mathcal{K}_l$. *Minimum energy efficiency maximization (MEEM) of the network, as given in (9), aims to balance the residual battery energy available across all the users at each time slot $l\in\{1,..,\mathsf{T}\}$ in the following manner:* To maximize $\min_{k\in\mathcal{K}_l} \ e'_k(l)/(E_k(l)+\mathcal{E}_k(l))$, energy consumption of an user with low residual battery energy would be low, and the energy consumption of an user with high residual battery energy would be high which is achieved by high computation and communication resource allocation for the user with low residual battery energy, and low communication and computation resource allocation for a user with high residual battery energy. Our experimental results in Section VI verify this induced property.

To allocate computation and communication resource allocation at each time slot $l\in\{1,..,\mathsf{T}\}$ according to MEEM, only task information for users in time slots l, $\beta_k(l), b_k(l)$, is required and therefore this strategy is easy to implement unlike (8). The optimal network lifetime problem in (8) aims to find the design variables that maximize network lifetime and therefore the network lifetime performance based on the solution of this strategy provides an upper bound to MEEM. Note that (8), or (9) may be infeasible if the value of T^{th} is very small. Next, we investigate solution methodologies for the problems (8), and (9).

IV. Minimum Energy Efficiency Maximization

Let V be a slack variable such that $1/V = \min_{k\in\mathcal{K}_l} \eta_k(l)$. Using (1)–(7), (9) can be expressed as

$$\begin{aligned}\min_{\boldsymbol{F'},\boldsymbol{B'},\boldsymbol{b'}} \quad & V, \\ \text{s.t.} \quad & \gamma_c\beta_k\left(b_k - b_k^{\text{EC}}\right) f_k^2 + P_k\frac{b_k^{\text{EC}}}{R_{k,b}} \le e'_k(l)V, \ \ k\in\mathcal{K}_l, \\ & \frac{\beta_k\left(b_k - b_k^{\text{EC}}\right)}{f_k} \le \mathsf{T}^{th}, \ \ k\in\mathcal{K}_l, \\ & \frac{\beta_k b_k^{\text{EC}}}{F_k} + \frac{b_k^{\text{EC}}}{B_k R_{k,b}} \le \mathsf{T}^{th}, \ \ k\in\mathcal{K}_l, \\ & \sum_{k\in\mathcal{K}_l} B_k \le B, \quad \sum_{k\in\mathcal{K}_l} F_k \le F. \end{aligned} \tag{10}$$

We omit the time slot index l above for notation brevity. The problem (10) is nonconvex since the third constraint is nonconvex. It can be converted to a geometric programming problem via the single condensation method [36]. According to this method, for a constraint which is a ratio of posynomials, the denominator posynomial (say $\mathrm{f}(\mathbf{x})$) can be approximated into a monomial using the following inequality:

$$\mathrm{f}(\mathbf{x}) = \sum_{\ell}\mathrm{f}_\ell(\mathbf{x}) \ge \hat{\mathrm{f}}(\mathbf{x}) = \prod_{\ell}\left[\frac{\mathrm{f}_\ell(\mathbf{x})}{\delta_\ell}\right]^{\delta_\ell}, \tag{11}$$

where $\delta_\ell > 0$ and $\sum_\ell \delta_\ell = 1$. Then, for $\delta_\ell = \mathrm{f}_\ell(\hat{\mathbf{x}})/\mathrm{f}(\hat{\mathbf{x}})$, $\hat{\mathrm{f}}(\hat{\mathbf{x}})$ is the best monomial approximation of $\mathrm{f}(\mathbf{x})$ near $\mathbf{x}=\hat{\mathbf{x}}$.

We formulate an iterative technique to optimally solve (10). At each iteration t, the first constraint in (10) is converted into a posynomial using (11) as

$$\left(\frac{e'_k(l)V(t)}{\delta_1(t)}\right)^{-\delta_1(t)}\left(\frac{\gamma_c\beta_k b_k^{\text{EC}}(t) f_k^2}{\delta_2(t)}\right)^{-\delta_2(t)} \cdot\left(\gamma_c\beta_k b_k f_k^2 + P_k\frac{b_k^{\text{EC}}(t)}{R_{k,b}}\right) \le 1, \quad k\in\mathcal{K}_l, \tag{12}$$

Algorithm 1: Algorithm for MEEM.

1: Set $t=1$, initialize $V(t)$, $F_k(t)$, $B_k(t)$, $b_k^{\mathrm{EC}}(t)$, $k \in \mathcal{K}_l$ such that the feasibility of (10) is preserved.
2: **while** true **do** ▷ infinite loop
3: $t = t+1$
4: Calculate $\delta_1(t)$, $\delta_2(t)$, $\delta_3(t)$ and $\delta_4(t)$
5: Find the optimum $V(t)$, $F_k(t)$, $B_k(t)$, $b_k^{\mathrm{EC}}(t)$, $k \in \mathcal{K}_l$ by solving (14) using GGPLAB [37]
6: **if** $|V(t)-V(t-1)| \leq \epsilon$ **then**
7: Break
8: **end if**
9: **end while**

where $\delta_1(t)$, and $\delta_2(t)$ are obtained from the solution at the $(t-1)$-th iteration as

$$\delta_1(t) = \frac{e'_k(l)V(t-1)}{e'_k(l)V(t-1)+\gamma_c\beta_k b_k^{\mathrm{EC}}(t-1)f_k^2},$$

$$\delta_2(t) = \frac{\gamma_c\beta_k b_k^{\mathrm{EC}}(t-1)f_k^2}{e'_k(l)V(t-1)+\gamma_c\beta_k b_k^{\mathrm{EC}}(t-1)f_k^2}.$$

Similarly, at each iteration t, the second constraints therein is converted into a posynomial using (11) as

$$\beta_k b_k \left(\frac{\mathsf{T}^{th} f_k}{\delta_3(t)}\right)^{-\delta_3(t)} \left(\frac{\beta_k b_k^{\mathrm{EC}}(t)}{\delta_4(t)}\right)^{-\delta_4(t)} \leq 1, \quad k \in \mathcal{K}_l, \quad (13)$$

where

$$\delta_3(t) = \frac{\mathsf{T}^{th} f_k}{\mathsf{T}^{th} f_k + \beta_k b_k^{\mathrm{EC}}(t-1)}, \delta_4(t) = \frac{\beta_k b_k^{\mathrm{EC}}(t-1)}{\mathsf{T}^{th} f_k + \beta_k b_k^{\mathrm{EC}}(t-1)}.$$

Thus, the overall optimization to be solved at iteration t is

$$\min_{\substack{V(t),F_k(t),B_k(t)\\ b_k^{\mathrm{EC}}(t),k\in\mathcal{K}_l}} \quad V(t)$$

$$\text{s.t.} \quad (12), \quad (13)$$

$$\frac{\beta_k b_k^{\mathrm{EC}}(t)}{F_k(t)} + \frac{b_k^{\mathrm{EC}}(t)}{B_k(t)R_{k,b}} \leq \mathsf{T}^{th}, \quad k \in \mathcal{K}_l$$

$$\sum_{k\in\mathcal{K}_l} B_k(t) \leq B, \quad \sum_{k\in\mathcal{K}_l} F_k(t) \leq F. \quad (14)$$

The above optimization problem is geometric programming and can be solved optimally. The iterative optimization is carried out until $|V(t)-V(t-1)| \leq \epsilon$ with $0 \leq \epsilon \ll 1$. An algorithmic implementation is included in Algorithm 1, which converges to the global solution of (10). The proof of the convergence of Algorithm 1 to the global solution of (10) available in [36].

Implementation Of MEEM: The resource allocation according to MEEM strategy can be implemented in a centralized manner. For this purpose, task information of the present time slot for all the users should be available at the BS which is similar to the centralized resource allocation strategies in literature [2], [3], [6]–[10]. Additionally, the BS should also have the residual energy information of the users to implement the resource allocation. We assume that information of the initial battery energy of the users is available at the BS, which can be obtained with a one-time transmission from the users. Then, the BS can calculate the energy consumption at each time slot and find the available residual energy for the next time slot.

Complexity Of Solution Strategy: Since CVX is used to solve GP sub-problems with the interior point method in step 5, the number of required iterations is $\frac{\log((3|\mathcal{K}_l|+2)/t_0\epsilon)}{\log\xi}$ where $|\mathcal{K}_l|$ is the number of active users at time slot l and hence $3|\mathcal{K}_l|+2$ is the total number of constraints, t_0 is the initial point to approximate the accuracy of interior point method, $0<\epsilon<1$ is the stopping criterion for interior point method, and ξ is used for updating the accuracy of interior point method [38]. For each iteration, the number of computations required to convert the non-convex problems into (12) and (13) is on the order of $|\mathcal{K}_l|$. Therefore, the total number of computations for Algorithm 1 is on the order of $|\mathcal{K}_l| \times \frac{\log((3|\mathcal{K}_l|+2)/t_0\epsilon)}{\log\xi}$.

Since we have considered ergodic data rate in (3), the proposed solution depends upon large-scale channel gain. If the users do not change their position significantly from a time slot to another and the task parameters do not change from a time slot to another, the resource allocation and data partition remain unchanged. Therefore, it is not necessary to run the proposed algorithms in each time slot.

V. Optimal Lifetime Maximization

Using (1)–(7), the problem in (8) can be expressed as

$$\max_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad \mathsf{T},$$

$$\text{s.t.} \quad \sum_{l\in S_k^{\mathsf{T}}} \left(\gamma_c\beta_k(l)\left(b_k(l)-b_k^{\mathrm{EC}}(l)\right)f_k^2 + P_k\frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}}\right) \leq e_k,$$

$$k \in \{1,..,K\},$$

$$\frac{\beta_i(m)\left(b_i(m)-b_i^{\mathrm{EC}}(m)\right)}{f_i} \leq \mathsf{T}^{th},$$

$$i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}\},$$

$$\frac{\beta_i(m)b_i^{\mathrm{EC}}(m)}{F_i(m)} + \frac{b_i^{\mathrm{EC}}(m)}{B_i(m)R_{i,b}} \leq \mathsf{T}^{th},$$

$$i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}\},$$

$$\sum_{i\in\mathcal{K}_m} B_i(m) \leq B, \quad \sum_{i\in\mathcal{K}_m} F_i(m) \leq F, \quad m \in \{1,..,\mathsf{T}\}. \quad (15)$$

Let $\mathsf{T}=\mathsf{T}'$ be a given value of T. The following feasibility test decides if the network will operate up to T' time slots:

$$\min_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad 0$$

$$\text{s.t.} \quad \sum_{l\in S_k^{\mathsf{T}'}} \left(\gamma_c\beta_k(l)\left(b_k(l)-b_k^{\mathrm{EC}}(l)\right)f_k^2 + P_k\frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}}\right) \leq e_k,$$

$$k \in \{1,..,K\},$$

$$\frac{\beta_i(m)\left(b_i(m)-b_i^{\mathrm{EC}}(m)\right)}{f_i} \leq \mathsf{T}^{th},$$

T T

$$
\begin{aligned}
& i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}, \\
& \frac{\beta_i(m) b_i^{\mathrm{EC}}(m)}{F_i(m)} + \frac{b_i^{\mathrm{EC}}(m)}{B_i(m) R_{i,b}} \leq \mathsf{T}^{th}, \\
& i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}, \\
& \sum_{i \in \mathcal{K}_m} B_i(m) \leq B, \quad \sum_{i \in \mathcal{K}_m} F_i(m) \leq F, \ m \in \{1,..,\mathsf{T}'\}
\end{aligned} \tag{16}
$$

Thus, problem (15) can be solved in a two-nested search loop in which we vary the value of T' in the outer loop, and in the inner loop, check if (16) is feasible. The maximum value of T', for which (16) is feasible, is the optimal network lifetime. We consider the following optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad & S, \\
\text{s.t.} \quad & \sum_{l \in S_k^{\mathsf{T}'}} \left(\gamma_c \beta_k(l) \left(b_k(l) - b_k^{\mathrm{EC}}(l) \right) f_k^2 + P_k \frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}} \right) \leq e_k, \\
& k \in \{1,..,K\}, && \text{(17a)} \\
& \frac{\beta_i(m) \left(b_i(m) - b_i^{\mathrm{EC}}(m) \right)}{f_i} \leq S, \\
& i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}, && \text{(17b)} \\
& \frac{\beta_i(m) b_i^{\mathrm{EC}}(m)}{F_i(m)} + \frac{b_i^{\mathrm{EC}}(m)}{B_i(m) R_{i,b}} \leq S, \\
& i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}, && \text{(17c)} \\
& \sum_{i \in \mathcal{K}_m} B_i(m) \leq B, \quad m \in \{1,..,\mathsf{T}'\}, && \text{(17d)} \\
& \sum_{i \in \mathcal{K}_m} F_i(m) \leq F, \quad m \in \{1,..,\mathsf{T}'\}, && \text{(17e)}
\end{aligned}
$$

Proposition 1: The feasibility testing in (16) can be solved in two steps, first to solve (17) optimally, and then check if the optimal value of S for T' time slots, $S_{T'}$ which is obtained by solving (17), is less than or equal to T^{th}.

Proof: See Appendix B. ■

Problem (17) can be converted into geometric programming, similarly to Section IV. We apply an iterative technique to solve it. At each iteration t, using (11), the first constraint in (17) is converted into a posynomial as

$$
\begin{aligned}
& \left(\frac{e_k}{\delta_5(t)} \right)^{-\delta_5(t)} \prod_{j \in S_k^{\mathsf{T}'}} \left(\frac{\gamma_c \beta_k(j) b_k^{\mathrm{EC}}(j,t) f_k^2}{\delta_{6j}(t)} \right)^{-\delta_{6j}(t)} \\
& \cdot \sum_{l \in S_k^{\mathsf{T}'}} \left(\gamma_c \beta_k(l) b_k(l) f_k^2 + P_k \frac{b_k^{\mathrm{EC}}(l,t)}{R_{k,b}} \right) \leq 1, \ k \in \{1,..,K\}
\end{aligned} \tag{18}
$$

where

$$
\delta_5(t) = \frac{e_k}{e_k + \sum_{l \in S_k^{\mathsf{T}'}} \gamma_c \beta_k(l) b_k^{\mathrm{EC}}(l,t-1) f_k^2},
$$

$$
\delta_{6j}(t) = \frac{\gamma_c \beta_k(j) b_k^{\mathrm{EC}}(j,t-1) f_k^2}{e_k + \sum_{l \in S_k^{\mathsf{T}'}} \gamma_c \beta_k(l) b_k^{\mathrm{EC}}(l,t-1) f_k^2},
$$

and the second constraint is converted into a posynomial as

$$
\begin{aligned}
\beta_i(m) b_i(m) \left(\frac{S(t) f_i}{\delta_9(t)} \right)^{-\delta_9(t)} \left(\frac{\beta_i(m) b_i^{\mathrm{EC}}(m,t)}{\delta_{10}(t)} \right)^{-\delta_{10}(t)} \leq 1, \\
i \in \mathcal{K}_m, \ \ m \in \{1,..,\mathsf{T}'\},
\end{aligned} \tag{19}
$$

where

$$
\delta_9(t) = \frac{S(t-1) f_i}{S(t-1) f_i + \beta_i(m) b_i^{\mathrm{EC}}(m,t-1)},
$$

$$
\delta_{10}(t) = \frac{\beta_i(m) b_i^{\mathrm{EC}}(m,t-1)}{S(t-1) f_i + \beta_i(m) b_i^{\mathrm{EC}}(m,t-1)}.
$$

Thus, the overall optimization to be solved at time t is:

$$
\begin{aligned}
\min_{\substack{S(t), F_i(m,t), \\ B_i(m,t), b_i^{\mathrm{EC}}(m,t)}} \quad & S(t), \\
\text{s.t.} \quad & (18), \ (19), \\
& \frac{\beta_i(m) b_i^{\mathrm{EC}}(m,t)}{F_i(m,t)} + \frac{b_i^{\mathrm{EC}}(m,t)}{B_i(m,t) R_{i,b}} \leq S, \\
& \qquad i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}, \\
& \sum_{i \in \mathcal{K}_m} B_i(m,t) \leq B, \quad m \in \{1,..,\mathsf{T}'\}. \\
& \sum_{i \in \mathcal{K}_m} F_i(m,t) \leq F, \quad m \in \{1,..,\mathsf{T}'\}.
\end{aligned} \tag{20}
$$

The above optimization is geometric programming and can be solved optimally. Hence, the optimal solution of (17) is obtained by solving (20) iteratively, following similar steps as given in Algorithm 1 [36]. Thus, to solve (15), in the inner loop, we check if the network operates for T' time slots, by first solving (17), following the approach as stated above, and then checking the condition $S_{T'} \leq \mathsf{T}^{th}$, for the given value of T'. In the outer loop, we then use the bisection search to find the maximum value of T' for which the network operates. The overall procedure is described in Algorithm 2. The output of the algorithm $T_{optimal}$ is the optimal network lifetime.

Even though this strategy may not be practically implementable due to its high computational complexity and the requirement for future user task information, it represents an upper bound for the performance of the MEEM approach. We show that in certain settings MEEM achieves the same performance as the globally optimal network lifetime strategy. The following proposition provides the upper bound of the optimal network lifetime when all the users have same task parameters.

Proposition 2: If each user k has the same task characteristics in every time slot, i.e., $\phi_k(l) = \phi_k = (\beta_k, b_k)$, the optimal network lifetime which can be obtained by solving (15) is upper bounded as follows

$$
T_{optimal} \leq min_{k \in \{1,..,K\}} \frac{e_k}{\epsilon_k}, \tag{21}
$$

Algorithm 2: Finding the Optimal Network Lifetime.

Initialize low and $high$ (lower and upper bounds for bisection search)
while $high > low$ **do**
  $T' = \lfloor \frac{low+high}{2} \rfloor$
  Find $S_{T'}$ by solving (17)
  if $S_{T'} < \mathsf{T}^{th}$ **then**
    $low = T' + 1$
  else
    $high = T'$
  end if
end while
$T_{optimal} = low - 1$

where

$$\epsilon_k = \begin{cases} \gamma_c \beta_k b_k f_k^2, & \text{if } \gamma_c \beta_k f_k^2 \leq \dfrac{P_k}{R_{k,b}}, \\ P_k \frac{b_k}{R_{k,b}}, & \text{if } \gamma_c \beta_k f_k^2 > \dfrac{P_k}{R_{k,b}}. \end{cases}$$

Proof: See Appendix B. ■

VI. Performance Evaluation

Here we present simulation results that evaluate the network lifetime performance of the proposed strategies. As a reference, we will compare our proposed strategies with the following benchmarks:

- **Reference Method 1** This scheme aims to minimize total energy consumption of all the users, i.e., minimize $\sum_{k \in \mathcal{K}_l} (E_k(l) + \mathcal{E}_k(l))$ at each time slot l with the same constraints of the problem of (9). Many state-of-the-art the works in literature [7], [8], [10], [11] have considered sum energy minimization as objective to decide resource allocation for mobile-edge computing networks.
- **Reference Method 2** This scheme aims to minimize the maximum energy consumption across all the users, i.e., minimize $\max_{k \in \mathcal{K}_l} (E_k(l) + \mathcal{E}_k(l))$ at each time slot l with the same constraints of the problem of (9). Minimizing maximum energy consumption across all the users at each time slot aims to provide fairness in energy consumption across the users to improve network lifetime.
- **Local Computation** In this scheme, the users compute the tasks at their own processors.
- **Full Offload** In this scheme, all the users are allocated equal computation and communication resources. At each time slot, all the bits of each user's task are offloaded to the edge cloud and computed at the processor of the edge cloud.

The resource allocation for reference methods 1 and 2 can be solved using geometric programming iteratively with similar steps as given in Algorithm 1. For the evaluations that follow, ten users are uniformly distributed in a circular region of radius 50 m with a cloud-associated BS at the center. The simulation parameters, unless mentioned otherwise, are summarized in Table II. Each user in the network is activated according to an activation probability p_i which follows the uniform distribution with $[0.3, 0.7]$. Therefore, the set of users which are activated at different time slots may be different. The obtained results are averaged over 500 network realizations.

TABLE II
Simulation Parameters

Parameter	Value
f_i	0.5 GHz [6, 9, 33]
β_i	Uniform in $[500, 1500]$ cycles/bit [6]
b_i	Uniform in $[100, 500]$ Kb [6, 9, 38]
P_i	10^{-8} W/Hz
B	5 MHz
F	6 GHz [9, 33]
T^{th}	0.15 s [9, 33]
γ_c	10^{-28} [6]
N_0	-147 dBM/Hz
ϵ	10^{-5}

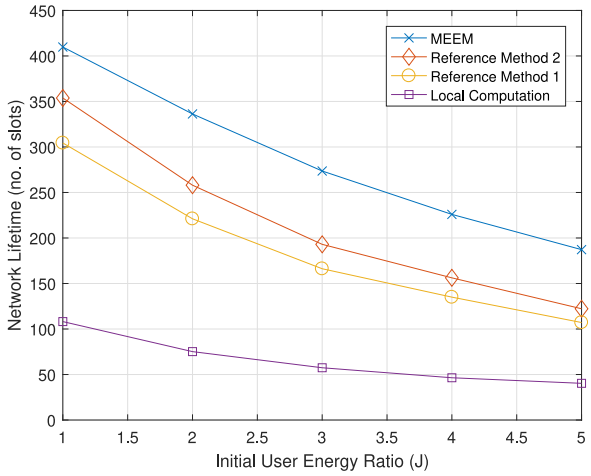

Fig. 3. Network lifetime versus e_{ratio} for total energy $e_{tot} = 5$ J. Local computation scheme does not meet the maximum tolerable delay per time slot and is included here only for illustration.

Note that the local computation and full offload schemes do not utilize all the resources available in the network and therefore for these two strategies, the users do not meet the maximum tolerable delay. The average delay achieved by local computation and full offload schemes are 0.21 s and 0.20 s, respectively, as indicated in the following figures.

In Fig. 3, we consider the performance of the proposed strategy in a network where the initial energy of the users is not identical. The network has a total of ten users among which five randomly chosen users have initial energy e_1 and the other five users have initial energy e_2 with $e_2 \leq e_1$. We fix the initial total network energy (i.e., the sum of battery energy of all users) as $e_{tot} = 5$ J. The network lifetime performance of the proposed strategies is evaluated when the initial user energy ratio $e_{ratio} = e_1/e_2$ varies from 1 to 5. If $e_{ratio} = 1$, we have identical initial energy for all the users, i.e., $e_1 = e_2 = 0.5$ J,

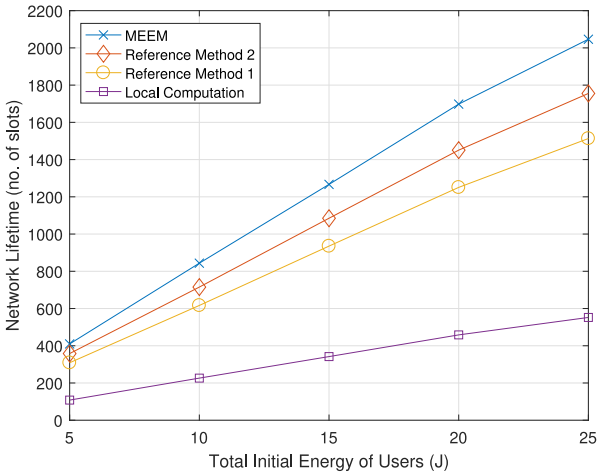

Fig. 4. Network lifetime versus total initial energy for the users. Local computation scheme does not meet the maximum tolerable delay per time slot and is included here only for illustration.

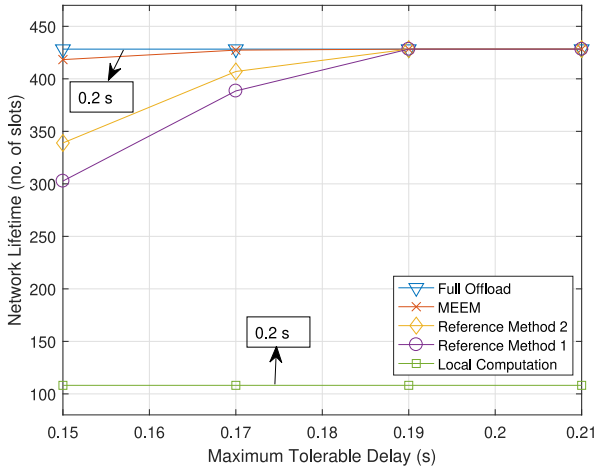

Fig. 5. Network lifetime versus T^{th} for total energy $e_{tot} = 5$ J. Local computation and full offload schemes do not meet the maximum tolerable delay per time slot and is included here only for illustration.

and if $e_{ratio} = 5$, five users have initial energy $e_1 = 0.83$ and the other five users have initial energy $e_2 = 0.17$. As e_{ratio} is increased, e_1 increases more compared to e_2, and the energy balancing decreases in the network. Thus, the network lifetime decreases for all strategies. Since the MEEM strategy considers the residual battery energy information to decide on the task sharing and resource allocation, while the reference methods do not consider the residual battery energy information to optimize the system parameters, MEEM achieves significant performance improvement compared to reference methods for high values of e_{ratio}. For example, if $e_{ratio} = 5$, the MEEM strategy achieves 1.73 times longer network lifetime (70% improvement) and 1.53 times longer network lifetime (50% improvement) compared reference methods 1 and 2, respectively. Furthermore, for $e_{ratio} = 5$, MEEM has 4.6 times higher network lifetime (460% improvement) compared to the local computation scheme.

Fig. 4 shows the network lifetime performance of the investigated strategies when the initial total network energy e_{tot} varies from 5 J to 25 J and $e_{ratio} = 1$. As e_{tot} increases, the network lifetime performance improves for each strategy. It can be observed that the rate of improvement of the MEEM and reference methods compared to the local computation strategies is higher. For $e_{tot} = 25$ J, the MEEM strategy achieves 1.15, 1.35 and 3.70 times longer network lifetimes compared to the reference methods 1, 2 and local computation respectively.

Fig. 5 shows the enabled network lifetime when the maximum tolerable delay T^{th} increases from 0.15 s to 0.21 s. We have $e_{ratio} = 1$, $e_{tot} = 5$ J. Here, we also show the performance of the full offload strategy. It can be observed that the full offload strategy has a higher network lifetime compared to the local computation strategy since the energy consumption for the users in offloading the task is lower compared to computing the task at their processor. However, none of these strategies is a practical choice, since the tasks can not be completed within the maximum tolerable delay for these strategies. As T^{th} increases, the network lifetime performance for MEEM and reference methods increases. This is because with an increase in the maximum tolerable delay, the decisions on sharing of tasks and allocation of computation and communication resources become more relaxed, and thus the energy consumption decreases for the users. Moreover, we can observe that for high values of T^{th}, MEEM has the same performance as reference methods and full offload, i.e., they converge. This is because energy consumption in offloading the task to the edge cloud for task computation is lower compared to the energy consumption in computing the task at local processor and at high value of T^{th}, each task is completed within maximum tolerable delay by offloading all the bits of each task at the edge cloud for MEEM and reference methods.

Next, we study the performance of MEEM and reference methods compared to the optimal network lifetime strategy described in Section V. For the latter, the number of optimization variables is proportional to the number of time slots over which the network operates. Thus, if the network lifetime is high, finding the optimal solution of (15) via geometric programming is challenging with many optimization variables. Hence, we show the performance of the proposed strategies when e_{tot} is low for which the network lifetime is low, and thus the number of optimization variables is small. In Fig. 6, we obtain the network lifetime performance of the proposed strategies for $e_{ratio} = 1$. We can observe that with local computation providing the worst performance, followed by reference methods 1, 2 and MEEM, enabling increasingly longer network lifetime, in the middle, and the Optimal Network Lifetime strategy providing the best performance, as expected. Similarly, we can observe that as e_{tot} increases, the network lifetime for all strategies increases, as expected, as well. The optimal network lifetime strategy achieves 12–17% higher network lifetime compared to MEEM, as e_{tot} varies, and that MEEM enables a consistent network lifetime gain of 21–45% relative to reference methods 1 and 2 respectively.

Fig. 7 compares the network lifetime enabled by the investigated strategies, as the computing power of the edge cloud varies from 5 to 9 GHz, for $e_{tot} = 0.5$ and $e_{ratio} = 1$. As the cloud

T T

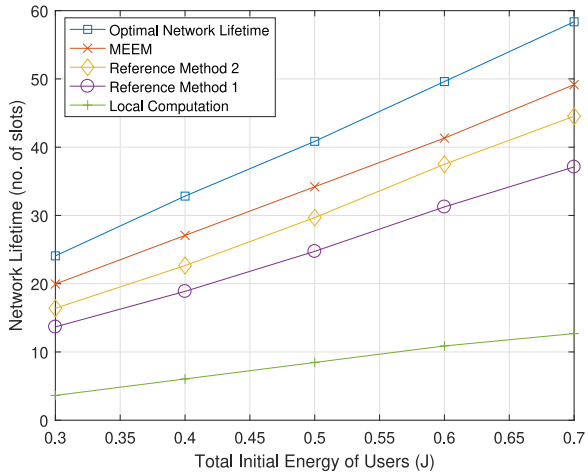

Fig. 6. Network lifetime versus total initial energy for the users. Local computation scheme does not meet the maximum tolerable delay per time slot and is included here only for illustration.

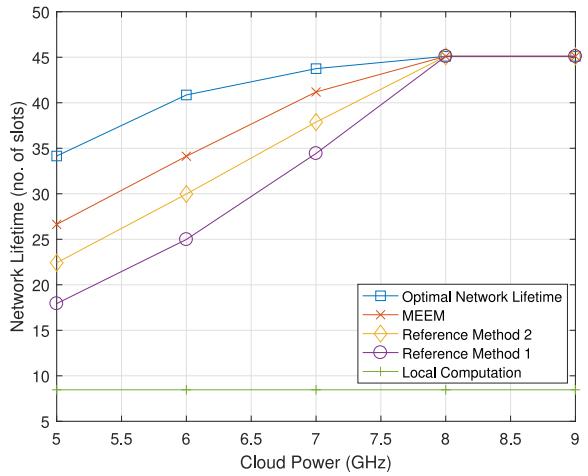

Fig. 7. Network lifetime versus cloud computing power. Local computation scheme does not meet the maximum tolerable delay per time slot and is included here only for illustration.

computing power increases, the network lifetime increases for all strategies. With an increase in cloud computing power, cloud computing power allocated to each user increases. Therefore, the users can offload more bits to the edge cloud, which helps to reduce the computation energy consumption for each user. This results in an improvement in the achieved network lifetime. Particularly, the users $k \in \mathcal{K}_l$ which have tasks with a high value of $\beta_k(l)$, can save high energy by offloading bits to the edge cloud and not computing at its own processor. Finally, for high cloud computing power, optimal network lifetime strategy, MEEM and reference methods have same performance, as observed from Fig. 7. This is because, at high value of cloud computing power, each user is allocated high computational resource of the edge cloud and the task for each user is completed within maximum tolerable delay by offloading all the bits of each task at the edge cloud to save energy consumption.

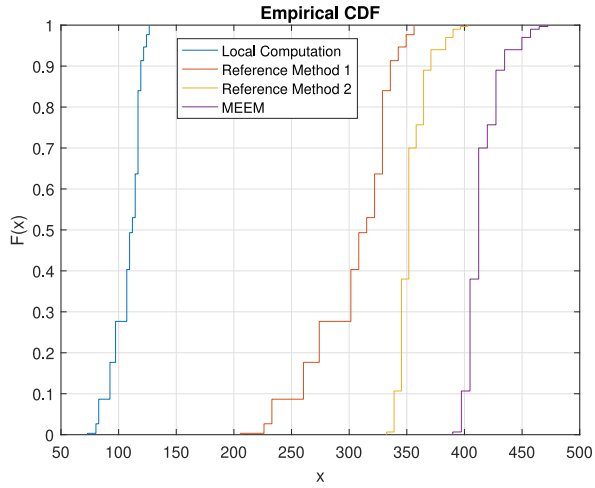

Fig. 8. Cumulative distribution function of enabled network lifetime.

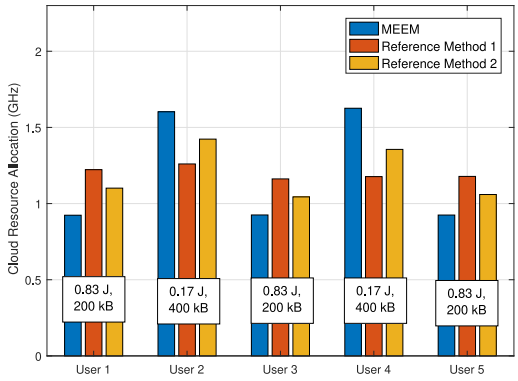

Fig. 9. Cloud computation power distribution among users.

In Fig. 8, we examine the empirical cumulative distribution function (CDF) of the network lifetime, achieved by each of the investigated strategies, for $e_{tot} = 5$ and $e_{ratio} = 1$. A total of 1000 network realizations have been considered in generating these results. The expected network lifetime achieved by local computation, reference methods 1, 2, and MEEM are 108.1, 304.3, 353.8 and 414 time slots, respectively. We note that these values match with the network lifetime performance demonstrated by these strategies in Fig. 5, for $e_{tot} = 5$ J. The standard deviation of the enabled network lifetime is 12, 34, 12.3, and 14 (in time slots), respectively, for local computation, reference methods 1, 2, and MEEM, respectively. Thus, MEEM considerably improves over reference method 1, not only in enabled expected network lifetime, but, also in its consistency across different network realizations. In particular, a close to three times reduction in network lifetime standard deviation is observed for MEEM compared to reference method 1 from Fig. 8.

In Figs. 9 and 10, we analyze the computation and communication resource distribution among the users at a given time slot. We consider that five users are active at the time slot and the initial battery energy of the users 1 to 5 are 0.83 J, 0.17 J, 0.83 J, 0.17 J and 0.83 J, respectively. Similarly, the number of bits to be computed by the users 1 to 5 are 200 Kb, 400 Kb,

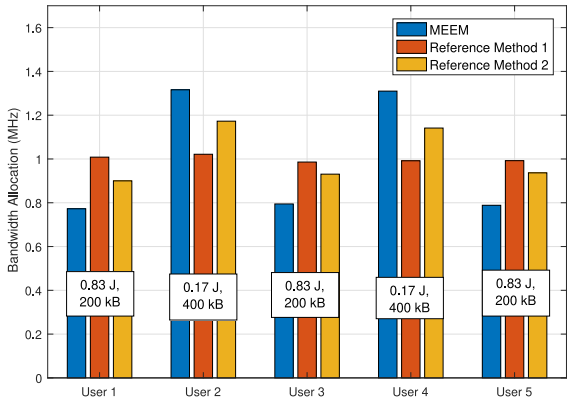

Fig. 10. Bandwidth distribution among users.

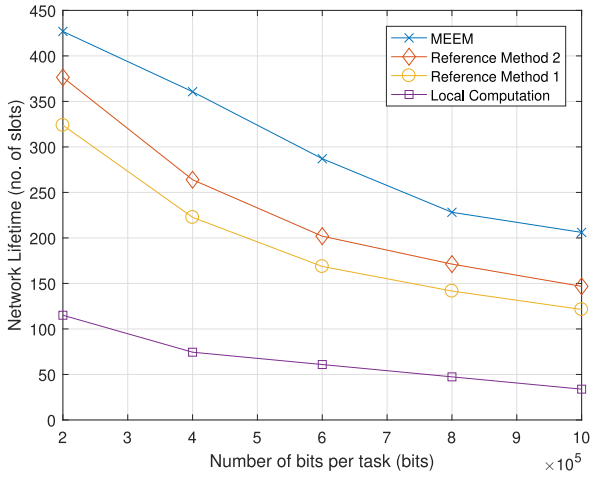

Fig. 11. Network lifetime versus b_i for total energy $e_{tot} = 5$ J. Local computation scheme does not meet the maximum tolerable delay per time slot and is included here only for illustration.

200 Kb, 400 Kb and 200 Kb, respectively. The initial battery energy and the number of computation bits for each user have been shown in rectangular box in the figures. The computation and communication resource allocation are balanced across the users for reference method 1. However, MEEM allocates higher computation and communication resources for the users with lower residual energy, larger number of computation bits, and it allocates lower computation and communication resources for the user with higher residual energy, smaller number of computation bits. Therefore, it performs better. Reference method 2 aims to minimize the maximum energy consumption across the users and therefore it allocates higher computation and communication resources for the users with larger number of computation bits and lower computation and communication resources for the user with smaller number of computation bits.

In Fig. 11, we examine the performance of the proposed strategies in terms of enabled network lifetime for a given number of bits b_i to be computed per task. The initial battery energy of the users is distributed randomly (uniform distribution) according to the values $e_{tot} = 5$ and $e_{ratio} = 1$. As b_i increases, the network lifetime performance decreases for each strategy. This is because

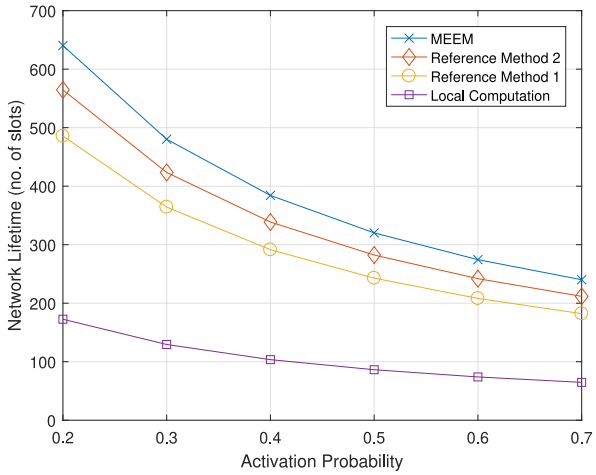

Fig. 12. Network lifetime versus activation probability.

higher values of b_i reflects that more bits need to be computed per task, which results in higher energy consumption at each time slot. It can be observed that the performance of the local computation strategy is poor for high values of b_i.

In Fig. 12, we analyze the network lifetime performance of the proposed strategies with the activation probability of the users in the network. For this purpose, we consider that all the users have the same activation probability and then we vary the activation probability of the users and observe network lifetime performance. As the activation probability increases, more users become active at each time slot, to compute their tasks, and therefore energy consumption increases. Thus network lifetime decreases with the increase in activation probability.

VII. Conclusion and Future Work

We investigated the lifetime maximization problem in a network where its nodes/users periodically compute their task with the help of an edge cloud. Aiming to maximize the network lifetime based on the user task information for the present time slot only, we have proposed an MEEM strategy to decide the sharing of tasks between the users and the cloud, and the allocation of computation and communication resources. We further investigated network lifetime maximization when future user task information is available, as well, as an upper bound to MEEM. Though the optimization problem for MEEM is non-convex, we have shown that the global optimal solution can be obtained using feasibility testing and geometric programming. We have shown that the MEEM strategy performs close to the optimal network lifetime. For high value of the initial user energy ratio, MEEM achieves roughly 70% lifetime improvement over the state-of-the-art and 460% lifetime improvement relative to local user computation only. For a high value of the maximum tolerable delay for completing the computation tasks of the users, MEEM achieves the globally optimal network lifetime performance. Finally, we have shown that MEEM achieves a significant reduction (3X) in variation of enabled network lifetime over diverse network topologies, compared to state-of-the-art.

In our approach, we considered quasi-static user mobility and a linear relationship between required CPU cycles and number of bits for a computing task. These assumptions are commonly encountered in practical settings [6], [7], [28]–[31] and enable analytical tractability and insightful results. Investigating dynamic user mobility within a computation offloading period and non-linear dependencies between required CPU cycles and task size in bits lie beyond the scope of the present paper and represent prospective avenues of future work. In particular, when the number of CPU cycles required for computation at the local device or the edge cloud can be expressed as a polynomial function of the task size in bits, a closed-form solution for the related resource allocation strategies can be obtained following the methods provided in Sections IV-V. Another prospective avenue of future work is to consider binary offloading of non-splittable tasks in our setting, where a task computation cannot be shared across the edge cloud and a user device. Finally, integrating our analytical advances into related emerging application settings, such as decentralized multi-view sensing, cooperative video streaming and caching, UAV-IoT, and mobile virtual reality [40]–[47] represents yet another prospective topic of future exploration.

APPENDIX A
PROOF OF PROPOSITION 1

Let $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ be a feasible solution point of (17), i.e., the constraints (17a), (17d) and (17e) are met at this point. If $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ is also a feasible solution of (16), then the value of

$$S = \max_{i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}} \left(\frac{\beta_i(m)\left(b_i(m) - b_i^{\mathrm{EC}}(m)\right)}{f_i}, \frac{\beta_i(m) b_i^{\mathrm{EC}}(m)}{F_i(m)} + \frac{b_i^{\mathrm{EC}}(m)}{B_i(m) R_{i,b}} \right)$$

is less than or equal to T^{th}. If $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ is not a feasible solution of (16), then $S > \mathsf{T}^{th}$. Therefore $S_{T'}$ must be less than or equal to T^{th} if there exist a feasible solution of (15).

APPENDIX B
PROOF OF PROPOSITION 2

To obtain the upper bound for the optimal network lifetime, as given in (15), we consider the following relaxed problem

$$\min_{\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b}} \quad 0,$$

$$\text{s.t.} \quad \sum_{l \in S_k^{\mathsf{T}'}} \left(\gamma_c \beta_k(l) \left(b_k(l) - b_k^{\mathrm{EC}}(l)\right) f_k^2 + P_k \frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}} \right) \leq e_k,$$

$$k \in \{1,..,K\}, \tag{22a}$$

$$\sum_{i \in \mathcal{K}_m} B_i(m) \leq B, \quad m \in \{1,..,\mathsf{T}'\}, \tag{22b}$$

$$\sum_{i \in \mathcal{K}_m} F_i(m) \leq F, \quad m \in \{1,..,\mathsf{T}'\}. \tag{22c}$$

The optimal network lifetime according to the above relaxed problem is an upper bound of the original problem (15). In case, the value of T^{th} is high, (15) becomes equivalent to (22). Since (22a), (22b), and (22c) are independent of each other, the problem of deciding the optimal share of the task to be offloaded for every user $k \in \{1,..,K\}$ reduces to this problem:

$$\min_{b_k^{\mathrm{EC}}(l)} \left(\gamma_c \beta_k(l) \left(b_k(l) - b_k^{\mathrm{EC}}(l)\right) f_k^2 + P_k \frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}} \right), \tag{23}$$

for each $l \in S_k^{\mathsf{T}'}$. The above problem is a linear optimization problem with a single variable and the optimal solution is

$$b_k^{\mathrm{EC}}(l) = \begin{cases} 0, & \text{if } \gamma_c \beta_k(l) f_k^2 \leq \dfrac{P_k}{R_{k,b}}, \\ b_k(l), & \text{if } \gamma_c \beta_k(l) f_k^2 > \dfrac{P_k}{R_{k,b}}. \end{cases} \tag{24}$$

Therefore, the energy consumption for each user k is

$$\epsilon_k(l) = \begin{cases} \gamma_c \beta_k(l) b_k(l) f_k^2, & \text{if } \gamma_c \beta_k(l) f_k^2 \leq \dfrac{P_k}{R_{k,b}}, \\ P_k \dfrac{b_k(l)}{R_{k,b}}, & \text{if } \gamma_c \beta_k(l) f_k^2 > \dfrac{P_k}{R_{k,b}}. \end{cases} \tag{25}$$

If each user $k \in \{1,..,K\}$ has same task characteristics in every time slot, i.e., $\phi_k = (\beta_k, b_k)$, the energy consumption of each user k is ϵ_k, which is obtained by replacing $\beta_k(l) = \beta_k$ and $b_k(l) = b_k$ in (25). Then, the network lifetime for each user k is e_k/ϵ_k. Therefore, the optimal network lifetime is upper bounded according to (21).

REFERENCES

[1] S. Gupta and J. Chakareski, "Geometric programming for lifetime maximization in mobile edge computing networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.

[2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper, vol. 11, 2015, pp. 1–16.

[3] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[4] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[5] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[6] H. Q. Le, H. Al-Shatri, and A. Klein, "Efficient resource allocation in mobile-edge computation offloading: Completion time minimization," in *Proc. IEEE Int. Symp. Info. Theory*, Jun. 2017, pp. 2513–2517.

[7] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[8] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[9] S. Gupta and A. Lozano, "Computation-bandwidth trading for mobile edge computing," in *Proc. Annu. IEEE Consum. Commun. Netw. Conf.*, Jan. 2019, pp. 1–6.

[10] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–858, Jan. 2019.

[11] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[12] Z. Zhou, S. Zhou, J.-H. Cui, and S. Cui, "Energy-efficient cooperative communication based on power control and selective single-relay in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3066–3078, Aug. 2008.

T T

[13] D. Wu, Y. Cai, L. Zhou, and J. Wang, "A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1190–1200, Mar. 2012.
[14] F. Ke, S. Feng, and H. Zhuang, "Relay selection and power allocation for cooperative network based on energy pricing," *IEEE Commun. Lett.*, vol. 14, no. 5, pp. 396–398, May 2010.
[15] W. J. Huang, Y. W. P. Hong, and C. C. J. Kuo, "Lifetime maximization for amplify-and-forward cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1800–1805, May 2008.
[16] H. Hui, S. Zhu, and G. Lv, "Relay selection for lifetime extension in amplify-and-forward cooperative networks," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.
[17] F. Zuo and M. Dong, "Prediction-based energy-aware relay cooperation for lifetime maximization," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 352–355, Aug. 2012.
[18] M. Hajiaghayi, M. Dong, and B. Liang, "Maximizing lifetime in relay cooperation through energy-aware power allocation," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4354–4366, Aug. 2010.
[19] L. Pang *et al.*, "Energy aware resource allocation for incremental AF-OFDM relaying," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1766–1769, Oct. 2015.
[20] T. Himsoon, W. P. Siriwongpairat, Z. Han, and K. J. R. Liu, "Lifetime maximization via cooperative nodes and relay deployment in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 2, pp. 306–317, Feb. 2007.
[21] S. Gupta and R. Bose, "Energy-aware relay selection and power allocation for multiple-user cooperative networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–7.
[22] S. Gupta and R. Bose, "Partner selection based on optimal power allocation for lifetime maximization in cooperative networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3088–3102, Apr. 2017.
[23] X. Yang, Z. Liu, and Y. Yang, "Minimization of weighted bandwidth and computation resources of fog servers under per-task delay constraint," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.
[24] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 137–12 151, Dec. 2018.
[25] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.
[26] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.
[27] F. Wang, J. Xu, and Z. Ding, "Multi-antenna noma for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2019.
[28] X. Hu, K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.
[29] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M. Zhou, "MEETS: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4076–4087, Oct. 2018.
[30] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-Enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.
[31] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile-edge computation offloading: Energy-efficient resource management," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7590–7605, Nov. 2018.
[32] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput.*, Jun. 2012, pp. 794–802.
[33] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, Jun. 2010, pp. 1–7.
[34] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.
[35] W. C. Y. Lee, "Estimate of channel capacity in Rayleigh fading environment," *IEEE Trans. Veh. Technol.*, vol. 39, no. 3, pp. 187–189, Aug. 1990.
[36] G. Xu, "Global optimization of signomial geometric programming problems," *Eur. J. Oper. Res.*, vol. 233, no. 3, pp. 500–510, 2014.
[37] GGPLAB: A simple MATLAB toolbox for geometric programming, May 2006. [Online]. Available: http://www.stanford.edu/boyd/ggplab/
[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.
[39] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE Symp. Comput. Commun.*, 2012, pp. 1–7.
[40] J. Chakareski and P. Frossard, "Distributed collaboration for enhanced sender-driven video streaming," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 858–870, Aug. 2008.
[41] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proc. ACM SIGCOMM Workshop Virtual Reality Augmented Reality Netw.*, Los Angeles, CA, USA, Aug. 2017, pp. 36–41.
[42] A. Khreishah and J. Chakareski, "Collaborative caching for multicell-coordinated systems," in *Proc. IEEE INFOCOM Workshop Commun. Netw. Techniques for Contemporary Video*, Hong Kong, China, Apr. 2015, pp. 257–262.
[43] N. Thomos, J. Chakareski, and P. Frossard, "Randomized network coding for UEP video delivery in overlay networks," in *Proc. IEEE Int'l Conf. Multimedia Expo*, New York City, NY, USA, Jun./Jul. 2009, pp. 730–733.
[44] J. Chakareski, V. Velisavljević, and V. Stanković, "User-action-driven view and rate scalable multiview video coding," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3473–3484, Sep. 2013.
[45] J. Chakareski, "UAV-IoT for next generation virtual reality," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5977–5990, Dec. 2019.
[46] J. Chakareski, "Uplink scheduling of visual sensors: When view popularity matters," *IEEE Trans. Commun.*, vol. 2, no. 63, pp. 510–519, Feb. 2015.
[47] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017, pp. 1–7.

Sabyasachi Gupta received the M.Tech degree from the National Institute of Technology (NIT) Durgapur, Durgapur, India, and the Ph.D. degree from the Indian Institute of Technology Delhi, New Delhi, India. He has held research positions with the University of Pompeu Fabra, Barcelona, and with the University of Alabama, Tuscaloosa. His research interests include resource allocation for wireless networks, mobile-edge computing networks, investigating application of optimization technique and graph theory for wireless communication, routing algorithms. He was the recipient of the Institute Gold Medal from NIT Durgapur in 2010.

Jacob Chakareski (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Rice University, Houston, TX, USA, and Stanford University, Stanford, CA, USA. He is currently an Associate Professor with the Ying Wu College of Computing, New Jersey Institute of Technology, where he leads the Laboratory for VR/AR Immersive Communication (LION). His research interests span networked virtual and augmented reality, UAV IoT sensing and networking, real-time reinforcement learning, 5G wireless edge computing/caching, ubiquitous immersive communication, and societal applications. He received the Adobe Data Science Faculty Research Award in 2017 and 2018, the Swiss NSF Career Award Ambizione (2009), the AFOSR Faculty Fellowship in 2016 and 2017, and Best/Fast-Track Paper Awards at IEEE ICC 2017/2018 and IEEE Globecom 2016. He is the organizer of the first NSF Visioning Workshop on networked VR/AR communications. He held research appointments with Microsoft, HP Labs, and EPFL, and served on the advisory board of Frame, Inc. His research is supported by the NSF, AFOSR, Adobe, Tencent Research, NVIDIA, and Microsoft.