Optimal Schedule of Mobile Edge Computing for Internet of Things Using Partial Information

Xinchen Lyu, Wei Ni, Hui Tian, Ren Ping Liu, Xin Wang, *Senior Member, IEEE*,
Georgios B. Giannakis, *Fellow, IEEE*, and Arogyaswami Paulraj, *Fellow, IEEE*

***Abstract*— Mobile edge computing is of particular interest to Internet of Things (IoT), where inexpensive simple devices can get complex tasks offloaded to and processed at powerful infrastructure. Scheduling is challenging due to stochastic task arrivals and wireless channels, congested air interface, and more prominently, prohibitive feedbacks from thousands of devices. In this paper, we generate asymptotically optimal schedules tolerant to out-of-date network knowledge, thereby relieving stringent requirements on feedbacks. A perturbed Lyapunov function is designed to stochastically maximize a network utility balancing throughput and fairness. A knapsack problem is solved per slot for the optimal schedule, provided up-to-date knowledge on the data and energy backlogs of all devices. The knapsack problem is relaxed to accommodate out-of-date network states. Encapsulating the optimal schedule under up-to-date network knowledge, the solution under partial out-of-date knowledge preserves asymptotic optimality, and allows devices to self-nominate for feedback. Corroborated by simulations, our approach is able to dramatically reduce feedbacks at no cost of optimality. The number of devices that need to feed back is reduced to less than 60 out of a total of 5000 IoT devices.**

***Index Terms*— Mobile edge computing, Internet of Things, partial information, Lyapunov optimization.**

I. Introduction

OFFLOADING and executing tasks of wireless terminal at the edge of wireless networks, mobile edge computing (MEC) is able to bridge the gap between the limited capability of individual devices and computationally demanding mobile applications [1], [2]. It provides an effective means to reduce the cost of wireless terminals without compromising functionalities and services. This is of particular interest to the future Internet of Things (IoT), where thousands of inexpensive, computationally incapable devices can be deployed within the coverage of a base station [3].

Different from cloud computing with dedicated high-speed wired connections, MEC has all devices operate in shared wireless media which feature unpredictable channel changes, and become increasingly congested with the increase of devices [4]. Optimal offloading schedules are obviously important to MEC, but yet to be established due to sophisticated stochasticity and changing availability of resources at the MEC server, random data arrival per device, and fluctuating wireless channels [5]. Myopically optimized schedules which fail to capture the stochasticity would be inefficient over longer time horizons [6]. Optimal schedules have also been hindered by a typical need for explicit knowledge on network states. In the case of IoT, thousands of devices could be required to feed back their states and channels, whenever the states or channels change [7]. This would incur prohibitive overhead, congest the channel, and jeopardize practicality.

Most existing works on MEC has overlooked the stochasticity or assumed predictability, and optimized the offloading schedules deterministically in the presence of full knowledge on network states. In [8]–[10], non-convex optimization problems were myopically formulated and solved in a centralized manner. In [11]–[13], game theoretic approaches or submodular optimization methods were taken to produce myopic schedules per slot in a decentralized fashion. In [14] and [15], Markov decision processes (MDP) were formulated to optimize the schedules under the assumption of a priori knowledge on the statistics of stochastic processes, and solved by dynamic programming. However, these statistics are typically unknown in prior, and dynamic programming also suffers from the "curse-of-dimensionality" and offers little scalability.

A few recent works have started to take the stochasticity and unpredictability into the consideration and optimize offloading schedules foresightedly by using stochastic optimization techniques, i.e., Lyapunov optimization [16]–[18]. However, these works encompassed on different network architectures from MEC, e.g., point-to-point link for single device [16] and multi-hop link in wireless sensor networks [17], [18], and their results are inapplicable to MEC with thousands of IoT devices to be offloaded.

Manuscript received April 1, 2017; revised September 12, 2017; accepted September 25, 2017. Date of publication October 9, 2017; date of current version December 1, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61471060 and Grant 61421061. *(Corresponding author: Hui Tian.)*

X. Lyu is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Global Big Data Technologies Center, University of Technology Sydney, Sydney, NSW 2007, Australia.

W. Ni is with the Digital Productivity and Service Flagship, CSIRO, Canberra, NSW 2122, Australia.

H. Tian is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: tianhui@bupt.edu.cn).

R. P. Liu is with the Global Big Data Technologies Center, University of Technology Sydney, Sydney, NSW 2007, Australia.

X. Wang is with the Key Laboratory for Information Science of Electromagnetic Waves (MoE), Department of Communication Science and Engineering, Fudan University, Shanghai 200433, China.

G. B. Giannakis is with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA.

A. Paulraj is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2017.2760186

This paper generates new asymptotically optimal offloading schedules, which are tolerant to partial out-of-date network knowledge and stochastically maximize a time-average network utility balancing system throughput and fairness. The contribution of this paper is multi-folded: (a) A perturbed Lyapunov technique is proposed to decouple time couplings of offloading schedule, energy intake, and data admission, and convert the stochastic optimization of MEC offloading schedules to deterministic optimizations over time slots without loss of optimality. An $[\mathcal{O}(V), \mathcal{O}(1/V)]$-tradeoff can be achieved between stability and utility, so that the optimality loss of the proposed approach (as compared to the offline optimum) asymptotically diminishes at the cost of increasing queue length; (b) Particularly, the deterministic optimization per slot is meticulously formulated to a linear relaxation of a knapsack problem. The breaking item of the problem, achieving the asymptotical optimality, is identified. Its lower bound in the presence of partial out-of-date information is derived as the threshold. As a result, only the devices meeting the threshold are designed to feed back per slot, hence preserving the asymptotic optimality with reduced feedbacks; (c) Extensive simulations show that our approach is able to substantially improve system utility by 26% and dramatically reduce feedbacks per slot by over 98%. Only less than 60 out of 5000 devices need to feed back per slot. The feedbacks can decline with an increasingly large number of devices, due to the increasingly prominent urgency of some devices.

The rest of this paper is organized as follows. In Section II, the system model is presented. In Section III, we elaborate the proposed stochastic online optimization of offloading schedules, provided up-to-date knowledge on network states at the MEC server. In Section IV, we extend the approach under out-of-date knowledge and derive the optimality preserving threshold for the self-nomination of devices. Simulation results are shown in Section V, followed by conclusions in Section VI.

II. System Model

The MEC system that we consider consists of a base station (BS) and N IoT devices, e.g., smart meters for supply monitoring, actuators for industrial automation, and sensors for smart home and healthcare.[1] Let $\mathbf{N} = \{1, 2, \ldots, N\}$ collect the indexes for N IoT devices. The system operates in a slotted structure, $t \in \mathbb{T} = \{0, 1, \ldots\}$ with slot length T. Equipped with energy harvesting capabilities, the IoT devices can be powered by renewable energy. An MEC server is deployed at the BS, and sensory data from the IoT devices can be offloaded through wireless links and processed at the MEC server. In the presence of other uplink traffic to the BS, the number of available subchannels for the IoT devices, denoted by $K(t)$, can vary between time slots. The notations used in this paper are summarized in Table I.

TABLE I

Definitions of Notations

Notation	Definition
$\mathbf{N}$	The set of IoT devices
T	The slot length
$K(t)$	The number of available subchannels during slot t
$c_i(t)$	The channel capacity of device i during slot t
$\tau_i(t)$	The transmission duration of device i within slot t
$d_i(t)$	The size of data offloaded from device i during slot t
$A_i(t)$	The size of data generated by device i at slot t
$a_i(t)$	The size of data admitted at device i within slot t
$e_i(t)$	The energy consumption of data offloading during slot t
$E_{\mathrm{H}}^i(t)$	The amount of energy harvested by device i at slot t
$e_{\mathrm{H}}^i(t)$	The amount of energy stored at device i during slot t
ξ_i	The CPU cycles for processing a task bit of device i
$F(t)$	The total available CPU cycles at slot t
$\overline{X}$	The time-average of any stochastic process $X(t)$
$\phi(\overline{\boldsymbol{a}})$	The system utility balancing throughput and fairness
$Q_i(t)$	The data queue backlog of device i at slot t
$E_i(t)$	The battery level of device i at slot t
$C(t)$	The task backlog in terms of CPU cycles at slot t
$G_i(t)$	The virtual queue backlog of device i at slot t
$\alpha_i(t)$	The unit profit of device i at slot t
$\widehat{X}_i(t)$	The value of X under partial out-of-date knowledge
$\underline{\alpha}_b(t)$	The feedback threshold at slot t

Independent and identical distributed (i.i.d.) flat block fading channels are assumed, i.e., the channels remain unchanged during a time slot and vary between slots [19], [20]. Let $c_i(t)$ denote the channel capacity in the uplink of device i at slot t. $c_i^{\min} \le c_i(t) \le c_i^{\max}$, where $c_i^{\min}$ and $c_i^{\max}$ are the minimum and maximum capacities of the link, respectively, given finite transmit powers of IoT devices.

Within a subchannel, the transmissions of multiple IoT devices can be accommodated via time-division multiple-access. Consider that an IoT device is inexpensive, simple and narrow-band [3], [7]. It can only access a subchannel at the same time. Let $\tau_i(t)$ denote the transmission duration of device i within slot t, i.e.,

$$0 \le \tau_i(t) \le T, \quad \forall i \in \mathbf{N}, \tag{1}$$

and the maximum amount of data offloaded from device i to the MEC server can be given by $\widehat{d_i}(t) = c_i(t)\tau_i(t)$. In some cases, a device can be assigned with different subchannels at different times during a slot. The narrow-band device can switch between subchannels to offload tasks [3].

We define an offloading schedule $\boldsymbol{\tau}(t) = \{\tau_1(t), \ldots, \tau_N(t)\}$ by collecting the schedules of all IoT devices during slot t. The total transmission time of the devices must not exceed the number of available subchannel times slot length, given as

$$\sum\nolimits_{i\in\mathbf{N}} \tau_i(t) \le K(t)T. \tag{2}$$

We define $\mathbb{D}(t) = \{\boldsymbol{\tau}(t) | 0 \le \tau_i(t) \le T, \sum_{i\in\mathbf{N}} \tau_i(t) \le K(t)T\}$ by collecting all the feasible offloading schedules at time slot t.

[1] The consideration of a single BS is elemental, and can be readily applied to the case of multiple BSs where each BS schedules the devices in its coverage independently as proposed in this paper. This is due to the typical layered architecture of cellular networks where inter-cell interference is mitigated through (fractional) frequency reuse among BSs.

Let $A_i(t)$ denote the arrival of sensory data at device i at slot t. $A_i(t)$ is an i.i.d. random process with the maximum $A_i^{\max}$. Device i may only be able to admit part of the arrived data, denoted by $a_i(t)$, i.e., $0 \leq a_i(t) \leq A_i(t)$, based on the availability of its buffer.

Let $Q_i(t)$ be the backlog of the data queue at device i, and it is updated along the time, as given by

$$Q_i(t+1) = [Q_i(t) - d_i(t)] + a_i(t), \tag{3}$$

where

$$d_i(t) = \min\{Q_i(t), c_i(t)\tau_i(t)\}, \tag{4}$$

since the device cannot offload more than what it has. The corresponding energy consumption of the device for offloading the data, $e_i(t)$, can be written as

$$e_i(t) = p_i \tau_i(t), \tag{5}$$

where p_i is the transmit power of device i.

The energy harvesting at an IoT device can also be modeled as a stochastic arrival process [16]–[18], [21]. Let $E_{\mathrm{H}}^i(t)$ denote the amount of renewable energy harvested by device i during time slot t. Assume that $E_{\mathrm{H}}^i(t)$ is i.i.d. with the maximum $E_{\mathrm{H}}^{i,\max}$. Consider a finite battery at the device. Device i can only store part of the newly harvested energy at each time slot t, denoted by $e_{\mathrm{H}}^i(t)$, where $0 \leq e_{\mathrm{H}}^i(t) \leq E_{\mathrm{H}}^i(t)$.

Let $E_i(t)$ denote the battery backlog of device i at slot t. It can be updated by

$$E_i(t+1) = [E_i(t) - e_i(t)] + e_{\mathrm{H}}^i(t), \tag{6}$$

where $e_i(t) \leq E_i(t)$, since the energy used for offloading data must not exceed the energy available in the battery.

Upon the receipt of the task from device i at time slot t, i.e., $d_i(t)$ in (4), the MEC server can process the task with $f_i(t) = \xi_i d_i(t)$ CPU cycles or put the task into the queue for later processing, where ξ_i is the number of CPU cycles required per task bit of device i for the task.

Let $C(t)$ be the required CPU cycles to process the task queued at the MEC server. $C(t)$ can be updated by

$$C(t+1) = \max\left[C(t) - F(t), 0\right] + \sum\nolimits_{i\in\mathbf{N}} f_i(t), \tag{7}$$

where $F(t)$ specifies the total available CPU cycles at slot t. In the presence of other concurrent services, $F(t)$ is stochastic with the maximum $F^{\max}$. $\sum_{i\in\mathbf{N}} f_i(t)$ gives the total of the newly offloaded tasks during time slot t.

III. Online Optimization of Offloading in MEC

Leveraging between throughput and fairness in MEC, the system utility can be defined as

$$\phi\ (\overline{\boldsymbol{a}}) = \sum\nolimits_{i\in\mathbf{N}} \log(1 + \overline{a_i}), \tag{8}$$

where $\overline{X} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[X(t)]$ defines the time-average of any stochastic process $X(t)$, and $\overline{\boldsymbol{a}} = \{\overline{a_1}, \overline{a_2}, \cdots, \overline{a_N}\}$ collects the time-average total of admitted data at all N devices. Particularly, $\phi\ (\overline{\boldsymbol{a}})$ is a concave logarithm function that is able to balance the network throughput and fairness by discouraging individual devices to consume excessive energy for little utility gains [22].

We can formulate $\mathbf{P}$ to maximize the system utility as

$$\begin{aligned}
\mathbf{P:}\quad & \max_{\boldsymbol{\tau}(t),\boldsymbol{e}_{\mathrm{H}}(t),\boldsymbol{a}(t)} \ \phi\ (\overline{\boldsymbol{a}}) \\
& \text{s.t. } \mathbf{C1:}\ \boldsymbol{\tau}(t) \in \mathbb{D}(t), \quad \forall t \in \mathbb{T} \\
& \quad \mathbf{C2:}\ 0 \leq e_{\mathrm{H}}^i(t) \leq E_{\mathrm{H}}^i(t), \quad \forall i \in \mathbf{N},\ t \in \mathbb{T} \\
& \quad \mathbf{C3:}\ 0 \leq a_i(t) \leq A_i(t), \quad \forall i \in \mathbf{N},\ t \in \mathbb{T} \\
& \quad \mathbf{C4:}\ e_i(t) \leq E_i(t), \quad \forall i \in \mathbf{N},\ t \in \mathbb{T} \\
& \quad \mathbf{C5:}\ \overline{\mathrm{C}},\ \ \overline{E_i} \text{ and } \overline{Q_i} < \infty, \ \forall i \in \mathbf{N},
\end{aligned} \tag{9}$$

where $\boldsymbol{\tau}(t)$, $\boldsymbol{e}_{\mathrm{H}}(t)$ and $\boldsymbol{a}(t)$ are the offloading schedule, energy intake, and data admission to be optimized, respectively. Constraints **C1** to **C4** are self-explanatory, as discussed in Section II. **C5** ensures the stability of all the queues. Particularly, if the queue backlogs are upper bounded, the total data admission $\sum_{i\in\mathbf{N}} \overline{a_i}$ is equal to the time-average system throughput in the long term [23].

Note that $c_i(t)$, $A_i(t)$, $E_{\mathrm{H}}^i(t)$ and $K(t)$ and computing resources $F(t)$ at the MEC server, are all time-varying with little predictability. An offline optimization of $\mathbf{P}$ would violate causality; whereas a myopic optimization per time slot would compromise the optimality in the long term, due to time coupling of variables in **C1** to **C4**.

A. Perturbed Lyapunov Optimization

Lemma 1: $\boldsymbol{P}$ can be equivalently reformulated as

$$\begin{aligned}
\boldsymbol{P1:}\quad & \max_{\boldsymbol{\tau}(t),\boldsymbol{e}_{\mathrm{H}}(t),\boldsymbol{a}(t),\boldsymbol{\gamma}(t)} \ \overline{\phi\ (\boldsymbol{\gamma})} \\
& \text{s.t. } \textit{C1-C5 and C6: } \overline{\gamma_i} \leq \overline{a_i} \\
& \quad \textit{C7: } 0 \leq \gamma_i(t) \leq A_i^{\max}, \quad \forall i \in \mathbf{N},\ t \in \mathbb{T},
\end{aligned} \tag{10}$$

where $\gamma(t)$ is the defined auxiliary variables.

Proof: See Appendix A. □

By defining a device-specific virtual queue, **C6** can be reformed with improved tractability, where the backlog of the virtue queue, denoted by $G_i(t)$, can be updated by the difference of $\gamma_i(t)$ and $a_i(t)$, as given by

$$G_i(t+1) = \max\{G_i(t) + \gamma_i(t) - a_i(t), 0\}. \tag{11}$$

Clearly, $G_i(t)$ is stable if and only if **C6** is satisfied [23]. We can replace **C6** with the stability of $G_i(t)$, and **P1** can be rewritten as

$$\begin{aligned}
\mathbf{P2:}\quad & \max_{\boldsymbol{\tau}(t),\boldsymbol{e}_{\mathrm{H}}(t),\boldsymbol{a}(t),\boldsymbol{\gamma}(t)} \ \overline{\phi\ (\boldsymbol{\gamma})} \\
& \text{s.t. } \mathbf{C1\text{-}C5} \text{ and } \mathbf{C7} \\
& \quad \mathbf{C8:}\ \overline{G_i} < \infty, \quad \forall i \in \mathbf{N}.
\end{aligned} \tag{12}$$

Stochastic optimization, more specifically, Lyapunov optimization, is able to eliminate time coupling of variables, and enable online decision-makings, while preserving asymptotic optimality [23]. A perturbed Lyapunov function of **P2** can be defined as

$$L(t) = \frac{1}{2}\left\{C(t)^2 + \sum_{i\in\mathbf{N}}\left[Q_i(t)^2 + (E_i(t) - \theta_i)^2 + G_i(t)^2\right]\right\}, \tag{13}$$

where $\boldsymbol{\theta} = \{\theta_i, \forall i \in \mathbf{N}\}$ collects the new perturbing weights (biases) to ensure sufficient energy available in the batteries for optimal schedules durations at any slot. Let $\Theta\ (t) = \{C(t), Q_i(t), E_i(t), G_i(t), \forall i \in \mathbf{N}\}$ collect all the backlogs at slot t. A drift-plus-penalty function can be defined by

$$\Delta_V(t) = \Delta\ (t) - V\mathbb{E}\left[\sum\nolimits_{i\in\mathbf{N}} \log(1+\gamma_i(t))|\Theta(t)\right], \quad (14)$$

where $\Delta\ (t) \overset{\Delta}{=} \mathbb{E}[L(t+1) - L(t)|\Theta\ (t)]$ is the conditional Lyapunov drift, i.e., the conditional expectation of the difference of the perturbed Lyapunov function between two consecutive time slots, and $V \geq 0$ is a predefined coefficient to tune the tradeoff between the system utility and queue stability.

Minimizing an upper bound of $\Delta_V(t)$ per slot t can prevent unbounded growths of backlogs, while maximizing the time-average conditional expectation of the system utility [23]. The upper bound can be given in the following lemma.

Lemma 2: For any queue backlogs and actions, $\Delta_V(t)$ is upper bounded by

$$\begin{aligned}\Delta_V(t) \leq D &- V\mathbb{E}\left[\sum\nolimits_{i\in N} \log\ (1+\gamma_i(t))|\Theta\ (t)\right]\\ &+ C(t)\mathbb{E}[\sum\nolimits_{i\in N} f_i(t) - F(t)|\Theta\ (t)]\\ &+ \sum\nolimits_{i\in N} Q_i(t)\mathbb{E}[a_i(t) - d_i(t)|\Theta\ (t)]\\ &+ \sum\nolimits_{i\in N} [E_i(t) - \theta_i]\mathbb{E}[e^i_{\mathrm{H}}(t) - e_i(t)|\Theta\ (t)]\\ &+ \sum\nolimits_{i\in N} G_i(t)\mathbb{E}[\gamma_i(t) - a_i(t)|\Theta(t)] \qquad (15)\end{aligned}$$

where

$$\begin{aligned}D = &\frac{1}{2}\sum\nolimits_{i\in N}\{(p_iT)^2 + (E^{i,\max}_{\mathrm{H}})^2 + (c^{\max}_iT)^2 + 3(A^{\max}_i)^2\}\\ &+\frac{1}{2}\{\sum\nolimits_{i\in N}(\xi_i c^{\max}_i T)^2 + (F^{\max})^2\} \qquad (16)\end{aligned}$$

Proof: See Appendix B. □

Exploiting Lemma 2, we reformulate **P2** to minimize (15), subject to the instantaneous constraints in **P2**, as follows:

$$\begin{aligned}\textbf{P3:}\quad &\max_{\boldsymbol{\tau}(t),\boldsymbol{e}_{\mathrm{H}}(t),\boldsymbol{a}(t),\boldsymbol{\gamma}(t)} f(\boldsymbol{\gamma}(t)) + g(\boldsymbol{\tau}(t)) - \eta(\boldsymbol{e}_{\mathrm{H}}(t)) - \mu(\boldsymbol{a}(t))\\ &\text{s.t. } \mathbf{C1\text{-}C4}\text{and}\mathbf{C7}, \qquad (17)\end{aligned}$$

where $F(t)$ is suppressed, since it is independent of the variables;

$$f(\boldsymbol{\gamma}(t)) = \sum\nolimits_{i\in\mathbf{N}}[V\log\ (1+\gamma_i(t)) - G_i(t)\gamma_i(t)]; \quad (18)$$

$$g(\boldsymbol{\tau}(t)) = \sum_{i\in\mathbf{N}}\{[Q_i(t) - C(t)\xi_i]d_i(t) + [E_i(t) - \theta_i]e_i(t)\}; \quad (19)$$

$$\eta(\boldsymbol{e}_{\mathrm{H}}(t)) = \sum\nolimits_{i\in\mathbf{N}}[E_i(t) - \theta_i]e^i_{\mathrm{H}}(t); \quad (20)$$

$$\mu\ (\boldsymbol{a}(t)) = \sum\nolimits_{i\in\mathbf{N}}[Q_i(t) - G_i(t)]a_i(t). \quad (21)$$

Let $\overline{\boldsymbol{a}^*}$ denote the optimal solution for **P3**, and ϕ^{opt} be the maximum system utility which can only be achieved offline under the assumption of full knowledge across all time slots, i.e., the optimum for **P**. As shown in the following theorem, the optimum for **P3**, $\phi\ (\overline{\boldsymbol{a}^*})$, converges to the optimum for **P**, ϕ^{opt}, as V increases.

*Theorem 1: The gap between the optimum for **P3**, $\phi\ (\overline{\boldsymbol{a}^*})$, and the optimum for **P**, ϕ^{opt}, is less than D/V, i.e.,*

$$\phi^{\text{opt}} - \phi(\overline{\boldsymbol{a}^*}) \leq D/V. \quad (22)$$

Proof: See Appendix C. □

Theorem 2: With $\overline{\boldsymbol{a}^*}$, the backlogs of all the queues in the system is strictly bounded at each time slot t, as given by

$$Q_i(t) \leq V/\ln 2 + 2A^{\max}_i; \quad (23)$$

$$E_i(t) \leq \theta_i + E^{i,\max}_{\mathrm{H}}; \quad (24)$$

$$C(t) \leq \beta + \sum\nolimits_{i\in N}\xi_i c^{\max}_i T; \quad (25)$$

where

$$\beta = \max_{i\in\mathbf{N}}\{\frac{V/\ln 2 + 2A^{\max}_i}{\xi_i} + \frac{E^{i,\max}_{\mathrm{H}} p_i}{c^{\min}_i\xi_i}\}. \quad (26)$$

Proof: See Appendix D. □

Theorem 3: The weight perturbation parameter θ_i can be adjusted by

$$\theta_i = [V/\ln 2 + 2A^{\max}_i]c^{\max}_i/p_i + p_iT, \quad (27)$$

such that, when a device is scheduled to offload at time slot t, there is enough energy in its battery backlog for optimal schedules, i.e., $E_i(t) \geq p_iT$.

Proof: See Appendix E. □

Theorems 1, 2 and 3 show an $[O(V), O(1/V)]$-tradeoff between the queue lengths (or in other words, the queueing delay according to Little's theorem [24]) and the optimality loss. Here, the optimality loss is the gap of system utility between the proposed approach and the causality-violating offline optimization of the offloading schedule. That is to say, by fine-tuning V, our approach allows the queueing delay to be adjustable at the cost of the optimal system utility.

B. Optimal Offloading Schedules

Note that $\boldsymbol{\tau}(t), \boldsymbol{e}_{\mathrm{H}}(t), \boldsymbol{a}(t)$ and $\boldsymbol{\gamma}(t)$ are decoupled in both the objective and constraints in **P3**, and therefore can be optimized separately as follows to achieve $\overline{\boldsymbol{a}^*}$.

1) Optimal Auxiliary Parameter: The subproblem of optimizing auxiliary variables can be written as

$$\begin{aligned}&\max_{\boldsymbol{\gamma}(t)}\ f(\boldsymbol{\gamma}(t))\\ &\text{s.t. } 0 \leq \gamma_i(t) \leq A^{\max}_i, \quad \forall i \in \mathbf{N}, \qquad (28)\end{aligned}$$

where $\gamma_i(t)$ is decoupled from $\gamma_j(t)$ for any $j \neq i$. Therefore, $\gamma_i(t)$ can be optimized independently by solving

$$\begin{aligned}&\max_{\gamma_i(t)} V\log(1+\gamma_i(t)) - G_i(t)\gamma_i(t)\\ &\text{s.t. } 0 \leq \gamma_i(t) \leq A^{\max}_i \qquad (29)\end{aligned}$$

The first-order partial derivation of $f(\boldsymbol{\gamma}(t))$ can be given by $\frac{\partial f}{\partial \gamma_i} = \frac{V}{(1+\gamma_i(t))\ln 2} - G_i(t)$. If $\frac{\partial f}{\partial \gamma_i}|_{\gamma_i(t)=0} \leq 0$, $f(\boldsymbol{\gamma}(t))$ decreases monotonically in the region $\gamma_i(t) \geq 0$ and the optimum $\gamma_i(t)^* = 0$. Otherwise, the optimum is either at the stationary point $\frac{\partial f}{\partial \gamma_i} = 0$, i.e., $\gamma_i(t)^* = \frac{V}{G_i(t)\ln 2} - 1$, or on the boundary, i.e., $\gamma_i(t)^* = A_i^{\max}$. Therefore,

$$\gamma_i(t)^* = \begin{cases} 0, & \text{if } \dfrac{V}{\ln 2} - G_i(t) \leq 0; \\ \min\{\dfrac{V}{G_i(t)\ln 2} - 1, A_i^{\max}\}, & \text{otherwise.} \end{cases} \tag{30}$$

2) Optimal Energy Intake: Decoupled from (17), the subproblem of optimizing energy intake can be written as

$$\begin{aligned} &\min_{\boldsymbol{e}_{\mathrm{H}}(t)} \sum\nolimits_{i\in\mathbf{N}}[E_i(t) - \theta_i]e_{\mathrm{H}}^i(t) \\ &\text{s.t. } 0 \leq e_{\mathrm{H}}^i(t) \leq E_{\mathrm{H}}^i(t), \quad \forall i \in \mathbf{N}, \end{aligned} \tag{31}$$

where the optimal solution can be readily given by

$$e_{\mathrm{H}}^i(t)^* = \begin{cases} 0, & \text{if } E_i(t) \geq \theta_i; \\ E_{\mathrm{H}}^i(t), & \text{otherwise.} \end{cases} \tag{32}$$

The devices consistently push their battery backlogs towards $\boldsymbol{\theta}$. As shown in Theorem 3, sufficient energy is available in the batteries for optimal schedules by judiciously designing $\boldsymbol{\theta}$.

3) Optimal Data Admission: The decoupled subproblem of optimizing data admission can be written as

$$\begin{aligned} &\min_{\boldsymbol{a}(t)} \sum\nolimits_{i\in\mathbf{N}}[Q_i(t) - G_i(t)]a_i(t) \\ &\text{s.t. } 0 \leq a_i(t) \leq A_i(t), \quad \forall i \in \mathbf{N}, \end{aligned} \tag{33}$$

where the optimal admission decision can be given by

$$a_i(t)^* = \begin{cases} 0, & \text{if } Q_i(t) \geq G_i(t); \\ A_i(t), & \text{otherwise.} \end{cases} \tag{34}$$

4) Optimal Offloading Schedule: $\boldsymbol{\tau}(t)$ is optimized to maximize $g(\boldsymbol{\tau}(t))$, while satisfying (1) and (2) at each time slot t. From (3) and (4), it is clear that device i should not transmit more data than $Q_i(t)$ at slot t, i.e., $\tau_i(t) \leq Q_i(t)/c_i(t)$. Besides, **C4** requires that the transmission energy cannot exceed the available energy in the battery backlog, i.e., $\tau_i(t) \leq E_i(t)/p_i$. Therefore, (1) can be tightened to

$$0 \leq \tau_i(t) \leq T_i, \quad \forall i \in \mathbf{N}, \tag{35}$$

where $T_i = \min\{Q_i(t)/c_i(t), E_i(t)/p_i, T\}$. By rearranging (19), $\boldsymbol{\tau}(t)$ can be optimized by

$$\begin{aligned} \mathbf{P4:} \quad &\max_{\boldsymbol{\tau}(t)} \sum\nolimits_{i\in\mathbf{N}} \alpha_i(t)\tau_i(t) \\ &\text{s.t. (2) and (35),} \end{aligned} \tag{36}$$

where

$$\alpha_i(t) = [Q_i(t) - C(t)\xi_i]c_i(t) + [E_i(t) - \theta_i]p_i. \tag{37}$$

By interpreting $\alpha_i(t)$ as the unit profit of an "item" i and $K(t)T$ as the capacity of a knapsack, **P4** becomes the linear relaxation of a knapsack problem. The optimal solution for linear relaxation knapsack can be found by selecting the "item" with higher unit profit to fulfill the knapsack capacity [25].

Without loss of generality, we assume the devices are sorted in the descending order of unit profit, i.e., $\alpha_i(t) \geq \alpha_j(t)$ for $i > j$. The optimal solution is $\tau_i = T_i$ if $\sum_{j=1}^{i} T_j \leq K(t)T$. Define the breaking item to be the first device that is not allocated its full transmission duration specified in (35) [25]. The index of the breaking item b can identified, as given by

$$b = \arg\min_i \sum\nolimits_{j=1}^{i} T_j > K(t)T. \tag{38}$$

As a result, the optimal offloading schedule can be given by

$$\tau_i(t)^* = \begin{cases} T_i, & \text{if } i < b; \\ K(t)T - \sum_{i=1}^{b-1} T_i, & \text{if } i = b; \\ 0, & \text{otherwise,} \end{cases} \tag{39}$$

which, involving ordering the devices against their unit profits, can be achieved by using the quicksort algorithm with a complexity of $\mathcal{O}(N \log N)$ [26].

Algorithm 1 Online Optimization of Offloading in MEC

```
At each device $i$:
    Acquire $Q_i(t)$, $E_i(t)$, $c_i(t)$, $A_i(t)$ and $E_{\mathrm{H}}^i(t)$
    Perform the optimal solutions given by (30), (32) and (34)
    Feed back $Q_i(t)$, $E_i(t)$ and $c_i(t)$ to the MEC server
    Update $G_i(t+1)$ based on (11)
At the MEC server:
    Acquire $K(t)$, $F(t)$ and $C(t)$
    Collect feedbacks from all the devices and evaluate $\alpha_i(t)$
    Schedule the offloading by (39)
```

Note that the optimal solutions in (30), (32) and (34) can be locally computed at individual IoT devices, since the queue backlogs, wireless channels, data arrivals and energy harvesting can be measured at the devices without the coordination of the MEC server. However, (39) requires the knowledge on all the IoT devices, i.e., $c_k(t)$, $Q_k(t)$, $E_k(t)$, and $G_k(t)$ for $k \in \mathbf{N}$. At each time slot t, the MEC server needs each IoT device to feed back its link capacity $c_k(t)$ and backlogs $Q_k(t)$, $E_k(t)$ and $G_k(t)$ to evaluate $\tau_i(t)^*$ in (39). We propose to distribute part of these optimizations (30), (32) and (34) at the IoT devices, and the rest (39) at the MEC server, as summarized in Algorithm 1.

IV. Optimization of Offloading Schedules Under Out-of-Date Backlogs

As discussed, the link capacity and backlogs of each IoT device are needed in the MEC server to evaluate (39) for optimal offloading schedules. Consider practical IoT, however, the number of IoT devices can be hundreds to thousands, while the number of available subchannels can be far less, i.e., $K(t) \ll N$. Getting each device to feed back as per slot could be infeasible due to explosive overhead, and also inefficient since the majority of the feedbacks would not contribute to the optimization of the schedule.

From (39), it is clear that the IoT devices with high unit profits are given priority to offload. Particularly, the optimality of (39) would not be compromised, if the devices with unit profits less than the breaking item, i.e., $\alpha_i(t) < \alpha_b(t)$, do not feed back. To this end, there is an opportunity arising to substantially reduce feedbacks at every slot.

In this section, we generalize the proposed online optimization with up-to-date knowledge on devices' backlogs at the MEC server, to the scenario where the knowledge is out-of-date at the MEC server from slot to slot. Lyapunov optimization under partial information and out-of-date knowledge is established, which captures the case of practical interest where only part of IoT devices feed back their backlogs. Using this, we derive a threshold for $\alpha_i(t)$, denoted by $\underline{\alpha}_b(t)$, which enables a small number of devices to self-nominate to feed back, while preserving the asymptotic optimality stated in Theorem 1. Following are the details.

From (37), the MEC server evaluates $\alpha_i(t)$ based on $C(t)$, $c_i(t)$, $Q_i(t)$ and $E_i(t)$, where $C(t)$ is known to the server whereas the other three parameters need device i to feed back. In the absence of up-to-date feedbacks from most devices, we define $\widehat{Q}_i(t)$ and $\widehat{E}_i(t)$ for the data and energy backlogs of device i, respectively, to capture the out-of-date knowledge at the MEC server.

The MEC server can update $\widehat{Q}_i(t)$ and $\widehat{E}_i(t)$, as follows. In the case that device i does not feed back at slot t, the MEC server keeps the out-of-date backlogs unchanged, i.e., $\widehat{Q}_i(t+1) = \widehat{Q}_i(t)$ and $\widehat{E}_i(t+1) = \widehat{E}_i(t)$. In the case that device i feeds back at slot t, the MEC server refreshes the backlogs up-to-date, as given by

$$\begin{aligned} \widehat{Q}_i(t+1) &= Q_i(t) - d_i(t) + a_i(t), \\ \widehat{E}_i(t+1) &= E_i(t) - e_i(t) + e_{\mathrm{H}}^i(t). \end{aligned} \tag{40}$$

From (3) and (6), we can see that these backlogs do not exceed the actual backlogs, i.e., $\widehat{Q}_i(t) \le Q_i(t)$ and $\widehat{E}_i(t) \le E_i(t)$ for $i \in \mathbf{N}$ and $t \in \mathbb{T}$. In the absence of feedback from device i, the channel capacity of the device can be replaced by the minimum capacity of the link $c_i^{\min}$.

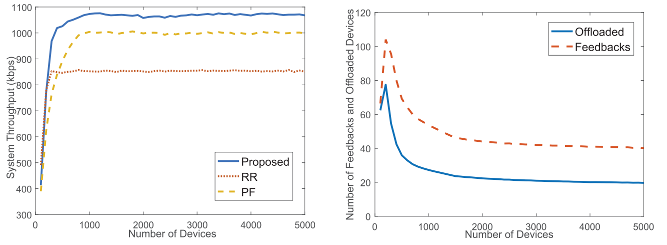
By referring to (37), the MEC server can evaluate the unit profit of device i based on the out-of-date backlogs, as given by

$$\widehat{\alpha}_i(t) = [\widehat{Q}_i(t) - C(t)\xi_i]c_i^{\min} + [\widehat{E}_i(t) - \theta_i]p_i. \tag{41}$$

Clearly, $\widehat{\alpha}_i(t) \le \alpha_i(t)$, $\forall i \in \mathbf{N}, t \in \mathbb{T}$. The unit profit of the breaking item $\underline{\alpha}_b(t)$ can be achieved by sorting devices in the descending order of $\widehat{\alpha}_i(t)$ and identifying the breaking item based on (38). Algorithm 1 can be slighted modified to accommodate the reduced feedbacks. Specifically, the MEC server broadcasts $\underline{\alpha}_b(t)$ as the threshold before step 1 based on its out-of-date information. Device i compares the threshold with its unit profit $\alpha_i(t)$, and self-nominates to feed back in step 3, if $\alpha_i(t) \ge \underline{\alpha}_b(t)$. Algorithm 1 resumes in the rest steps.

Corollary 1: The out-of-date knowledge on the backlogs of devices does not compromise the optimality of ***P3****.*

Proof: The optimal $\boldsymbol{e}_{\mathrm{H}}(t)$, $\boldsymbol{a}(t)$ and $\boldsymbol{\gamma}(t)$ for **P3** are obtained based on the real-time measurement and calculation at each device. We only need to prove the optimality of $\boldsymbol{\tau}(t)$ for **P4**. Since $\underline{\alpha}_b(t) \le \alpha_b(t)$ always holds, the optimal set of devices to be selected in (39) to offload, i.e., $\{i|\alpha_i(t) \ge \alpha_b(t)\}$, is a subset of the devices feeding back their link capacity and up-to-date backlogs, i.e., $\{i|\alpha_i(t) \ge \underline{\alpha}_b(t)\}$. This preserves the optimality of $\boldsymbol{\tau}(t)$. □



(a) System throughput (b) Feedbacks and offloaded devices

Fig. 1. The changes of throughput and the corresponding average number of device as N increases from 100 to 5000 where $V = 40$.

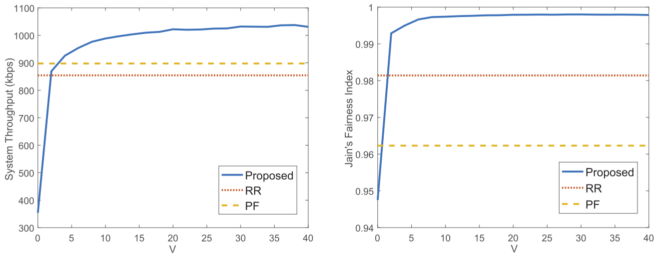
V. Simulation Results

In this section, we evaluate the proposed optimal offloading schedules of MEC system in Matlab with $T = 1$ sec. Focused on scheduling a large number of devices using partial out-of-date knowledge, we assume that the uploaded tasks can be computed correctly, and the randomness of externalities, such as sensory data, energy arrival and channel conditions, can be parameterized. Particularly, the computational resource $F(t) \sim U[0, 3]$ GHz, the arrival of sensory data $A_i(t) \sim U[0, 10]$ kbps, the number of available subchannels $K(t) \sim U[0, 30]$, where $U[a, b]$ denotes a random uniform distribution within $[a, b]$. We also assume that the arrival of renewable energy $E_{\mathrm{H}}^i(t) \sim U[0, 20]$ mJ/s, the transmit power $p_i \sim U[10, 23]$ dBm, and the average link capacity $c_i \sim U[20, 80]$ kbps, where $c_i(t) \sim U[0.5\ c_i, 2\ c_i]$. The number of devices N is set to be 500, unless otherwise specified. Every result of the simulations is the average of 5000 independent runs.

For comparison purpose, we also simulate (a) Round Robin (RR) method which gives no priority to any devices; and (b) Proportional Fair (PF) based on the popular weighted fair queueing model, where devices are prioritized and scheduled based on $c_i(t)/\sum_{\tau=0}^{t-1} d_i(t)$. We note that, in the case of RR, devices are scheduled in a predetermined order and only a subset of devices to be scheduled at the upcoming slot need to feed back their states on channels, battery levels and queue backlogs. In the case of PF, however, each device needs to feed back its priority and states, based on which the MEC server selects the devices to offload. In other words, PF still requires explicit up-to-date knowledge on all the devices.

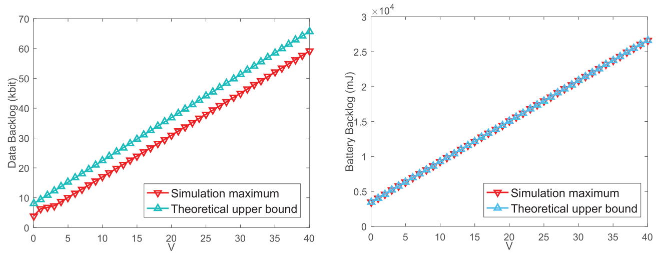
Fig. 1 plots the throughput of the proposed approach and the corresponding average number of devices that feed back per slot, where N increases from 100 to 5000. We can see in Fig. 1(a) that the throughput of the proposed approach converges as the number of devices gets large, e.g., $N > 1000$. This is due to the limited wireless link capacity and computational resources of the network. Nevertheless, our approach provides higher throughput than RR and PF, by prudently leveraging the data queues and energy battery.

More importantly, we can see in Fig. 1(b) that the number of devices self-nominated to feed back their backlogs and channels is substantially reduced in the proposed approach based on partial and out-of-date information. Particularly, the number of self-nominating devices is about twice the number of devices to be optimally scheduled. It is interest-



(a) System throughput (b) Fairness among devices

Fig. 2. The system throughput and Jain's fairness index of the proposed approach, RR and PF, where V ranges from 0 to 40 and $N = 500$.



(a) Data backlog (b) Battery backlog

Fig. 3. The comparison of the simulation results on the maximum backlogs of the data and energy queues where V increases from 0 to 40, where $N = 500$.

ing to see in Fig. 1(b) that there is a surge of throughput around $N = 200$, after which the number of self-nominating devices declines with the increasing number of devices. The reason is that the total data arrival of all devices is low and likely to be instantly offloaded in the case of $N < 200$. In contrast, the total data arrival is so high in the case of $N > 200$, and the data queues build up at the devices. At every slot, the urgency of a small number of devices for offloading tasks becomes increasingly prominent, leading to a growth of $\underline{\alpha}_b(t)$ and subsequent decrease of self-nominating devices.

Fig. 2 plots the system throughput and fairness of the proposed approach, as V increases [27]. Here, $V \in [0, 40]$ is displayed to provide a reasonable dynamic range to show the impact of V on both the queue length and throughput. We can see that both the throughput and Jain's index increase rapidly with V when $V \leq 5$, and then slow down increasing and start to stabilize when $V \geq 30$. As revealed in Theorem 1, the reason for this is that increasing V leads the utility to grow, which leverages both the throughput and fairness. For comparison, we also plot the throughput and Jain's index of RR and PF, both of which are independent of V and therefore remain unchanged. We can see that the proposed scheme approach is able to outperform RR and PF by 18% in terms of throughput and up to 4.5% in terms of Jain's index; see $V = 40$.

Fig. 3 compares the simulation results on the maximum backlogs of the data and energy queues where V increases from 0 to 40. We can see that the maximum backlog of battery tightly fits its upper bound given in (15), while the maximum backlog of data is consistently lower than its upper bound by about 5 kbits. This corroborates the upper bounds and the theorem – Theorem 2. The consistent differences between the backlogs and upper bounds also provide opportunities to predict the value for V and the expected maximum utility by using the upper bounds (15), given the sizes of data buffer and battery at the devices.

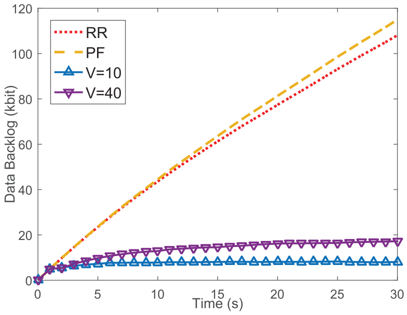
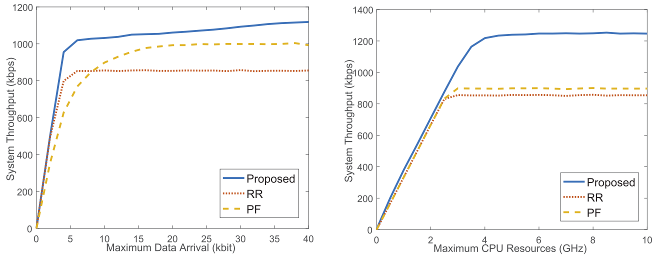


Fig. 4. The time variation of the average data backlog of the proposed approach, RR and PF along the time, where $N = 500$.



(a) Throughput versus data arrival, where $F_{\max} = 3$ GHz (b) Throughput versus CPU resources, where $A_i^{\max} = 10$ kbps

Fig. 5. The comparison of throughput between the proposed approach, RR and PF, versus data arrival and CPU resources, where $N = 500$ and $V = 40$.

Fig. 4 plots the time variation of the data backlog of the proposed approach in comparison with those of RR and RF along the time. We can see that the average data backlogs are able to quickly stabilize, as the time elapses. Moreover, a small value of V is able to further speed up the stabilization and also reduce the backlog, as compared to a large V value. This finding is consistent with Fig. 3. In Fig. 4, we also see that the average backlogs of both RR and PF continue increasing almost linearly to the time, and have yet to stable by the end of the simulation time. The superiority of the proposed approach in terms of system stability is confirmed.

Fig. 5 compares the throughput between the proposed approach, RR and PF, where the data arrival at the devices and the available computing resources at the MEC server are varied separately. We show in Fig. 5(a) that the throughput of the proposed approach increases with the growth of data arrival. Particularly, the throughput grows quickly and then starts to stabilize when $A_i^{\max} \geq 5$ kbps. This is because the data arrival of $A_i^{\max} = 5$ kbps starts to congest the network, and to saturate the offloading throughput. RR and PF also experience the congestion of the network and the saturation of the offloading throughput. However, the saturated throughput of RR and PF is much lower than that of the proposed approach. We can also see the network congestion and throughput saturation in Fig. 5(b), where the computing resources are increasingly available at the MEC server and the offloading throughput is restrained by the wireless link capacity of the network.

VI. Conclusion

In this paper, Lyapunov optimization techniques were taken to generate asymptotically optimal offloading schedules for MEC under partial network knowledge. Particularly, we decomposed the Lyapunov optimization problem into a knapsack problem which can be solved for asymptotically optimal schedules, given up-to-date network knowledge. We proceeded to evaluate the solution for the knapsack problem under out-of-date knowledge, and prove that the solution, encapsulating the asymptotically optimal schedules, preserves the asymptotic optimality. It can be used for devices to self-nominate for feedback and facilitate scheduling. Simulations show that the proposed approach is able to reduce feedbacks by over 98% without compromising the asymptotic optimality, e.g., only require 60 out of 5000 devices to feed back per slot.

Appendix A

Given that **C1** to **C5** are part of **P1**. We only need to prove that the maximum for **P1** is no less than that for **P** in the presence of **C6** and **C7**. Let $\overline{\phi\ (\boldsymbol{\gamma})}$ denote the maximum utility of **P1**, where $\overline{\boldsymbol{a}^*}$ is the corresponding time-average data admission. Since $\phi(\boldsymbol{x})$ is a monotonically increasing function, **C6** guarantees that $\phi\ (\overline{\boldsymbol{a}^*}) \geq \phi\ (\overline{\boldsymbol{\gamma}})$. Using Jensen's inequality [28], we have $\phi(\overline{\boldsymbol{\gamma}}) \geq \overline{\phi\ (\boldsymbol{\gamma})}$ and, in turn, $\phi\ (\overline{\boldsymbol{a}^*}) \geq \overline{\phi\ (\boldsymbol{\gamma})}$. Let ϕ^{opt} denote the solution for **P**, where $\boldsymbol{a}^{\text{opt}}(t)$ is the corresponding data admission at slot t. Clearly, under **C6** and **C7** in **P1**, performing the strategy $\boldsymbol{\gamma}(t) = \overline{\boldsymbol{a}^{\text{opt}}(t)}$ for all time slots t, achieves $\overline{\phi\ (\boldsymbol{\gamma})} = \phi^{\text{opt}}$. As a result, $\phi\ (\overline{\boldsymbol{a}^*}) \geq \phi^{\text{opt}}$, i.e., **P1** is equivalent to **P**.

Appendix B

Taking squares on both sides of (3), (7), (6) and (11), and then exploiting $(\max[a-b,0]+c)^2 \leq a^2+b^2+c^2+2a(c-b)$ for any $a, b, c \geq 0$, we have:

$$Q_i(t+1)^2 - Q_i(t)^2 \leq a_i(t)^2 + d_i(t)^2 + 2Q_i(t)[a_i(t) - d_i(t)], \tag{42}$$

$$C(t+1)^2 - C(t)^2 \leq (\sum\nolimits_{k\in\mathbf{N}} f_k(t))^2 + F(t)^2 + 2\ C(t)[\sum\nolimits_{k\in\mathbf{N}} f_k(t) - F(t)], \tag{43}$$

$$E_i(t+1)^2 - E_i(t)^2 \leq e_{\text{H}}^i(t)^2 + e_i(t)^2 + 2[E_i(t) - \theta_i][e_{\text{H}}^i(t) - e_i(t)], \tag{44}$$

$$G_i(t+1)^2 - G_i(t)^2 \leq a_i(t)^2 + \gamma_i(t)^2 + 2G_i(t)[\gamma_i(t) - a_i(t)]. \tag{45}$$

Plugging (42)-(45) into (14) yields the upper bound in Lemma 2 with D satisfying

$$D \geq \frac{1}{2}\sum_{i\in N}\{\mathbb{E}[e_i(t)^2] + \mathbb{E}[e_{\text{H}}^i(t)^2] + \mathbb{E}[\gamma_i(t)^2] + 2\mathbb{E}[a_i(t)^2] + \mathbb{E}[d_i(t)^2]\} + \frac{1}{2}\{\mathbb{E}[(\sum\nolimits_{i\in N} f_i(t))^2] + \mathbb{E}[F(t)^2]\}. \tag{46}$$

Then, (16) provides an upper bound for (46) by replacing all the expectations with the maximums in (46).

Appendix C

Let π denote any feasible control strategy for **P**. The minimum upper bound of $\Delta_V(t)$ in (15) can be given by

$$\begin{aligned}\Delta_V(t) \leq D &- V\mathbb{E}\left[\sum\nolimits_{i\in N} \log\ (1+\gamma_i(t))|\Theta\ (t), \pi\right] \\ &+ C(t)\mathbb{E}[\sum\nolimits_{i\in N} f_i(t) - F(t)|\Theta\ (t), \pi] \\ &+ \sum\nolimits_{i\in N} Q_i(t)\mathbb{E}[a_i(t) - d_i(t)|\Theta\ (t), \pi] \\ &+ \sum\nolimits_{i\in N} [E_i(t) - \theta_i]\mathbb{E}[e_{\text{H}}^i(t) - e_i(t)|\Theta\ (t), \pi] \\ &+ \sum\nolimits_{i\in N} G_i(t)\mathbb{E}[\gamma_i(t) - a_i(t)|\Theta(t), \pi].\end{aligned} \tag{47}$$

According to [23], for any $\sigma > 0$, there exists at least one randomized stationary control policy π^* (i.e., the policy is independent from the backlogs), such that:

$$\begin{aligned}-\phi\ (\boldsymbol{\gamma}(t)|\pi^*) &\leq -\phi^{\text{opt}} + \sigma, \\ \mathbb{E}[\sum\nolimits_{i\in N} f_i(t) - F(t)|\pi^*] &\leq \sigma, \\ \mathbb{E}[a_i(t) - d_i(t)|\pi^*] &\leq \sigma, \\ \mathbb{E}[e_{\text{H}}^i(t) - e_i(t)|\pi^*] &\leq \sigma, \\ \mathbb{E}[\gamma_i(t) - a_i(t)|\pi^*] &\leq \sigma.\end{aligned} \tag{48}$$

Plugging (48) into (47) and then taking $\sigma \to 0$ yields

$$\Delta_V(t) \leq D - V\phi^{\text{opt}}. \tag{49}$$

Taking expectation on both sides of (49), we obtain

$$\mathbb{E}[L(t+1) - L(t)] - V\mathbb{E}[\phi(\gamma\ (t))] \leq D - V\phi^{\text{opt}}. \tag{50}$$

Summing up all the telescoping series of (50) over $\{0, 1, \cdots, T-1\}$, we can obtain

$$\mathbb{E}[L(T)] - \mathbb{E}[L(0)] - V\sum\nolimits_{t=0}^{T-1}\mathbb{E}[\phi(\gamma\ (t))] \leq DT - VT\phi^{\text{opt}}. \tag{51}$$

Consider that all queues are empty initially. Given the fact that $\mathbb{E}[L(T)] \geq 0$ and $L(0) = \sum_{i\in N}\theta_i^2$, we have

$$\frac{1}{T}\lim_{T\to\infty}\sum\nolimits_{t=0}^{T-1}\mathbb{E}[\phi(\gamma\ (t))] \geq \phi^{\text{opt}} - \frac{D}{V}, \tag{52}$$

i.e., $\overline{\phi(\gamma\ (t))} \geq \phi^{\text{opt}} - \frac{D}{V}$. In Lemma 1, we have proved that $\phi\ (\overline{\boldsymbol{a}^*}) \geq \overline{\phi(\gamma\ (t))}$. Therefore, we obtain $\phi^{\text{opt}} - \phi\ (\overline{\boldsymbol{a}^*}) \leq \frac{D}{V}$.

Appendix D

We first prove the upper bound of the auxiliary variable backlogs, i.e., $G_i(t) \leq \frac{V}{\ln 2} + A_i^{\max}$ through mathematical induction. Specifically, the upper bound holds at slot $t = 0$ given the fact that $G_i(t) = 0, \forall i \in \mathbf{N}$. Suppose that the upper bound holds at time slot t. Then, if $G_i(t) \geq \frac{V}{\ln 2}$, $\gamma_i(t) = 0$ according to (30), and $G_i(t)$ cannot increase at slot t, i.e., $G_i(t+1) \leq G_i(t)$. Otherwise, if $G_i(t) < \frac{V}{\ln 2}$, the difference between $G_i(t)$ and $G_i(t+1)$ is less than $A_i^{\max}$ according to (11) and **C7**, i.e., $G_i(t+1) \leq \frac{V}{\ln 2} + A_i^{\max}$. As a result, the upper bound also holds at slot $t+1$. This concludes the proof of the upper bound of $G_i(t)$.

Given the upper bound of $G_i(t)$, we prove (23). Since (23) holds for $t = 0$, we suppose that it holds at slot t. From (34), if $Q_i(t) \geq G_i(t)$, device i does not admit new data, and therefore, $Q_i(t+1) \leq Q_i(t) \leq G_i(t) + A_i^{\max}$. On the other hand, if $Q_i(t) < G_i(t)$, the increased backlog must not exceed $A_i^{\max}$, i.e., $Q_i(t+1) \leq G_i(t) + A_i^{\max}$. Substituting the upper bound of $G_i(t)$, we have $Q_i(t+1) \leq \frac{V}{\ln 2} + 2\ A_i^{\max}$. In other words, (23) holds at slot $t+1$. By mathematical induction, (23) is proved.

Next, we proceed to prove (24). Since (24) holds at slot $t = 0$, we suppose that it holds at slot t. If $E_i(t) \geq \theta_i$, device i must not harvest energy according to (32), i.e., $E_i(t+1) \leq E_i(t) \leq \theta_i + E_{\mathrm{H}}^{i,\max}$. Otherwise, if $E_i(t) < \theta_i$, we have $E_i(t+1) \leq \theta_i + E_{\mathrm{H}}^{i,\max}$ due to $e_{\mathrm{H}}^i \leq E_{\mathrm{H}}^{i,\max}$. In other words, (24) holds at slot $t+1$. This concludes the proof of (24).

Finally, we prove (25). Since it holds at slot $t = 0$, we suppose that it holds at slot t. If $C(t) \geq \beta$, we have $\alpha_i(t) \leq 0$ for any device $i \in \mathbf{N}$, i.e., the MEC server does not schedule any device according to (39), resulting in $C(t+1) \leq C(t)$. If $C(t) < \beta$, we have $C(t+1) \leq \beta + \sum_{i \in N} \xi_i c_i^{\max} T$ by substituting the maximum link capacity. As a result, (25) holds at $t+1$. This concludes the proof of (25).

APPENDIX E

When θ_i is adjusted according to (27) and $E_i(t) < p_i T$ at slot t, we have $E_i(t) - \theta_i < -\frac{[V/\ln 2 + 2\ A_i^{\max}]c_i^{\max}}{p_i}$. Plugging this into (37), we obtain $\alpha_i(t) < [Q_i(t) - C(t)\xi_i]c_i(t) - [V/\ln 2 + 2\ A_i^{\max}]c_i^{\max} < Q_i(t)c_i(t) - [V/\ln 2 + 2\ A_i^{\max}]c_i^{\max}$, i.e., $\alpha_i(t) < 0$ from (23). This proves that a device is not scheduled if $E_i(t) < p_i T$, given θ_i in (27).

REFERENCES

[1] X. Lyu and H. Tian, "Adaptive receding horizon offloading strategy under dynamic environment," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 878–881, May 2016.

[2] W. Zhang, Y. Wen, J. Wu, and H. Li, "Toward a unified elastic computing platform for smartphones with cloud support," *IEEE Netw.*, vol. 27, no. 5, pp. 34–40, Sep./Oct. 2013.

[3] M. R. Palattella *et al.*, "Internet of Things in the 5G era: Enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, to be published.

[5] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[6] X. Chen, W. Ni, T. Chen, I. B. Collings, X. Wang, and G. B. Guanacos, "Real-time energy trading and future planning for fifth generation wireless communications," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 24–30, 2017.

[7] A. Rico-Alvarino *et al.*, "An overview of 3GPP enhancements on machine to machine communications," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 14–21, Jun. 2016.

[8] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, to be published

[9] C. Qin, W. Ni, H. Tian, R. P. Liu, and Y. J. Guo, "Joint beamforming and user selection in multiuser collaborative MIMO SWIPT systems with non-negligible circuit energy consumption," *IEEE Trans. Veh. Technol.*, to be published.

[10] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *Proc. WiMob*, Oct. 2015, pp. 794–801.

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

[12] 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.

[13] 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.

[14] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1326–1336, Apr. 2010.

[15] P. Blasco, D. Gunduz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.

[16] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[17] L. Huang and M. J. Neely, "Utility optimal scheduling in energy-harvesting networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1117–1130, Aug. 2013.

[18] X. Wang, T. Ma, R. Zhang, and X. Zhou, "Stochastic online control for energy-harvesting wireless networks with battery imperfections," *IEEE Trans. Wireless Commun.*, vol. 15, no. 12, pp. 8437–8448, Dec. 2016.

[19] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar, "An online learning algorithm for energy efficient delay constrained scheduling over a fading channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 732–742, May 2008.

[20] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[21] X. Chen, W. Ni, X. Wang, and Y. Sun, "Optimal quality-of-service scheduling for energy-harvesting powered wireless communications," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3269–3280, May 2016.

[22] S. Lin, W. Ni, H. Tian, and R. P. Liu, "An evolutionary game theoretic framework for femtocell radio resource management," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6365–6376, Nov. 2015.

[23] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan and Claypool Publishers, 2010.

[24] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*. San Diego, CA, USA: Academic, 2014.

[25] D. Pisinger, *Algorithms for Knapsack Problems*. Copenhagen, Denmark: Univ. Copenhagen, 1995.

[26] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. New York, NY, USA: McGraw-Hill, 2001.

[27] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," *SIGMETRICS Perform. Eval. Rev.*, vol. 28, no. 1, pp. 118–119, Jun. 2000.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Xinchen Lyu received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently pursuing the Cotutelle Ph.D. degree with BUPT and the University of Technology Sydney. His research interests include mobile edge computing and radio resource management.

Wei Ni received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is currently a Senior Scientist, and the Team and Project Leader at CSIRO, Australia. He also holds honorary positions at the University of New South Wales, Macquarie University, and the University of Technology Sydney. Prior to this, he was a Post-Doctoral Research Fellow with Shanghai Jiaotong University (2005–2008), a Research Scientist and the Deputy Project Manager at the Bell Labs R&I Center, Alcatel/Alcatel-Lucent (2005–2008), and a Senior Researcher at Devices R&D, Nokia (2008–2009). His research interests include optimization, game theory, graph theory, and their applications to network and security.

Dr. Ni has been serving as an Editor for the *Hindawi Journal of Engineering* since 2012, a Secretary for the IEEE NSW VTS Chapter since 2015, the Track Chair for VTC-Spring 2016 and 2017, and the Publication Chair for BodyNet 2015. He served as the Student Travel Grant Chair for WPMC 2014, a Program Committee Member for CHINACOM 2014, and a TPC Member of the IEEE ICC14, ICCC15, EICE14, and WCNC10.

Hui Tian received the M.S. degree in microelectronics and the Ph.D. degree in circuits and systems from the Beijing University of Posts and Telecommunications (BUPT), China, in 1992 and 2003, respectively. She is currently a Professor with BUPT, the Network Information Processing Research Center Director of the State Key Laboratory of Networking and Switching Technology, and the MAT Director of the Wireless Technology Innovation Institute. Her current research interests mainly include radio resource management, cross layer optimization, M2M, cooperative communication, mobile social network, and mobile edge computing.

Ren Ping Liu (SM'13) is currently a Professor with the School of Electrical and Data Engineering, University of Technology Sydney, where he leads the Network Security Laboratory, Global Big Data Technologies Centre. He is also the Research Program Leader of Digital Agrifood Technologies in Food Agility CRC, a government/research/industry initiative to empower Australia's food industry through digital transformation. Prior to that, he was a Principal Scientist with CSIRO, where he led wireless networking research activities. He specializes in protocol design and modeling, and has delivered networking solutions to a number of government agencies and industry customers. He has over 100 research publications, and has supervised over 30 Ph.D. students. His research interests include Markov analysis and QoS scheduling in WLAN, VANET, IoT, LTE, 5G, SDN, and network security. He received the Australian Engineering Innovation Award and the CSIRO Chairman Medal.

He received the B.E. (Hons.) and M.E. degrees from the Beijing University of Posts and Telecommunications, China, and the Ph.D. degree from The University of Newcastle, Australia. He served as the technical program committee chair and the organizing committee chair for a number of IEEE conferences. He is the Founding Chair of the IEEE NSW VTS Chapter.

Xin Wang received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree from Auburn University, Auburn, AL, USA, in 2004, all in electrical engineering.

He was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, from 2004 to 2006. In 2006, he joined the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA, where he is currently an Associate Professor. He is also a Distinguished Professor with the Department of Communication Science and Engineering, Fudan University. His research interests include stochastic network optimization, energy-efficient communications, cross-layer design, and signal processing for communications.

Georgios B. Giannakis (F'97) received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1981, and the M.Sc. degree in electrical engineering and mathematics and Ph.D. degree in electrical engineering from the University of Southern California, in 1983 and 1986, respectively. He was with the University of Virginia from 1987 to 1998. Since 1999, he has been a Professor with the University of Minnesota, where he was an Endowed Chair in wireless telecommunications, a University of Minnesota McKnight Presidential Chair in ECE, and the Director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing subjects on which he has authored over 400 journal papers, 680 conference papers, 25 book chapters, two edited books and two research monographs (h-index 119). Current research focuses on learning from big data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-)inventor of 28 patents issued. He is a fellow of the EURASIP. He was a (co-)recipient of eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received the Technical Achievement Award from the SP Society (2000) and EURASIP (2005), the Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.

Arogyaswami Paulraj (F'15) received the B.E. degree from the Naval College of Engineering, Lonavala, India, in 1966, and the Ph.D. degree from IIT Delhi, New Delhi, India, in 1973. He is a Pioneer of MIMO wireless communications. He joined Stanford University, Stanford, CA, in 1972, after a 20-year industry/military career in India. He has authored over 400 research publications and two text books, and is a co-inventor in 66 U.S. patents. He is a member of eight national academies, including those of USA and China. His recognitions include the 2014 Marconi Prize and the 2011 IEEE Alexander Graham Bell medal.