Adaptive Cooperative Task Offloading for Energy-Efficient Small Cell MEC Networks

Zewei Jing*, Qinghai Yang*, Yan Wu*, Meng Qin†, Kyung Sup Kwak‡, and Xianbin Wang§
*School of Telecommunication Engineering, Xidian University, Xi'an, China
†Shenzhen Pengcheng Laboratory, Shenzhen, China
‡Graduate School of Information Technology and Telecommunications, Inha University, Incheon, South Korea
§Department of Electrical and Computer Engineering, Western University, London, Canada

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771874

***Abstract*—Cooperative task offloading has emerged as a compelling computing paradigm for balancing spatially uneven task workloads and computational resources in distributed mobile edge computing (MEC) systems. However, enabling cooperation among multiple MEC nodes inevitably requires extra communication and computational energy overheads which might counteract the cooperation gain without energy-efficient offloading mechanisms. This paper presents an adaptive cooperative task offloading algorithm aiming at maximizing the time-averaged energy efficiency for small cell MEC networks enabled by millimeter-wave backhauls. With the considered network dynamics, the proposed algorithm makes a good tradeoff between the harvested cooperation utility and the total energy consumption in the long term. In addition, our algorithm ensures the network stability and fulfills the task admission rate requirement of each individual user equipment, by making slot-based decisions over time without requiring a-priori knowledge of the network dynamics. Simulation results verify the outstanding performance of the proposed algorithm by comparing with the static cooperative and adaptive non-cooperative schemes.**

***Index Terms*—Mobile edge computing, small cell, energy efficiency, cooperative task offloading.**

I. Introduction

As an important supplement to mobile cloud computing, mobile edge computing (MEC) has emerged as one of the key solutions to fulfill the stringent low-latency requirement of many burgeoning computation-intensive applications, such as augmented/virtual reality, interactive gaming, and so on [1]. In addition, millimeter-wave (mmWave) backhaul enabled small cell (SC) network has become one of the promising 5G technologies to enhance the spectrum and energy efficiencies per unit area owing to its cost-effectiveness and large available bandwidth. Therefore, it has become a recent trend to integrate the two critical technologies together by deploying a small and resource-limited edge server (ES) with each base station (BS) in the SC network [2]. By offloading the computation-intensive tasks of mobile user equipments (UEs) to the proximal ESs, the task completion latency can be remarkably reduced compared to the remote cloud offloading. Nevertheless, since the processing capabilities of ESs are quite limited due to the scarce onboard resources (e.g., CPU, memory), it could be impossible to provide satisfactory offloading experience to each UE by a signal ES, especially when the task traffic is extremely high in a certain region. Therefore, cooperative task offloading has drawn growing attention to balance the spatially uneven task workloads among multiple geographically distributed ESs [3]-[6]. Cooperative task offloading allows ESs to either compute the UE tasks by themselves locally or offload them further to other ESs through the connected backhaul links. In this way, both the resource utilization and the task processing rate can be enhanced at the same time.

However, the complicated network environment of the mmWave backhaul enabled SC-MEC network also introduces several unique technical challenges for cooperative task offloading. One major challenge is the energy-efficient design of the cooperative mechanism, since reaping benefits from cooperation inevitably incurs extra communication and computational energy overheads. The most widely adopted energy efficiency (EE) criterion is defined as the ratio of the overall throughput/utility to the total energy consumption, so that the EE maximization problem often conforms to a non-convex and intractable fractional programming problem. Another main technical challenge is the network variability, which primarily results from the stochastic UE task arrivals and time-varying mmWave backhauls. Due to the unpredictability of network dynamics, offline policies (e.g., dynamic programming) become inapplicable. Myopic optimization which overlooks the network variability would be inefficient in the long run. Moreover, UEs usually have different offloading experience requirements (e.g., task admission rate) depending on their application/task patterns. Simply allocating resources by treating all UEs equally would fail to guarantee their individual offloading experience requirements. Last but not least, the joint stochastic optimization of multi-dimensional communication and computational resources introduces severe variable couplings, which further deteriorates the problem tractability.

So far, most of existing works are based on myopic and offline optimizations. For instance, myopic optimization had been done for maximizing the cooperative task offloading social utility by using a two-tier matching game model in [3]. A collaborative cloud and edge computing framework was proposed in [4], where the total task completion latency was minimized with full network knowledge. The most related works to this paper which take the network variability into account are [5] and [6]. One-hop dynamic peer offloading was proposed in [5] for SC-MEC networks with wired backhauls. A tit-for-tat based incentive mechanism was proposed in [6] for multi-hop cooperative task offloading in fog computing.

However, [5] and [6] did not consider the energy-efficient design, the diverse UE offloading experience requirements as well as the stochastic impact of mmWave backhauls.

In this paper, we investigate the adaptive cooperative task offloading for EE maximization in SC-MEC networks with network variability. Different from [5], we exploit multi-hop wireless mmWave backhauls, which enable UE tasks to be offloaded among BSs/ESs over multi-hop paths. The EE objective is defined as the ratio of the time-averaged network utility which captures UE proportional fairness, to the time-averaged total energy consumption which involves the ES computation energy consumption and the BS backhaul communication energy consumption. Queue stability theory is exploited to compress the network queue backlog, as well as to meet the requirement of time-averaged task admission rate of each individual UE. We propose an Adaptive Cooperative Task Offloading algorithm, called ACTO, based on the Lyapunov optimization framework [1] and the fractional programming theory [7]. The proposed algorithm jointly optimizes the UE task admission, BS transmit power and transmit rate allocation, and the ES CPU-cycle allocation at each time slot without requiring a-priori knowledge of the network dynamics.

II. System Model and Problem Formulation

A. Network Model

We consider a discrete-time SC-MEC network which has one macro BS and M small BSs that are connected with each other by mmWave backhauls. The operational timeline of the system is divided into back-to-back time slots of same slot length τ, which is indexed by $t \in \{0, 1, 2, \ldots\}$. We assume that the mmWave backhaul link between any pair of BSs operates in a full-duplex transmission mode with perfect self-interference cancellation capability [8]. The network communication topology is characterized by a directed graph $\mathcal{G} = (\mathcal{M}, \mathcal{L})$, where $\mathcal{M} = \{0, 1, \ldots, M\}$ collects all BS nodes and index 0 specifically refers to the macro BS. $\mathcal{L} = \{(i, j)|i, j \in \mathcal{M}\}$ represents the set of all backhaul links (i, j) where BS i and j are the transmitter and the receiver, respectively. We gather all the receiving BSs $j \in \mathcal{M}$ of the backhaul links with BS i as the transmitter by set $\mathcal{M}_i$. It is obvious that under the assumption of full-duplex transmission, we will have BS $i \in \mathcal{M}_j$ if BS $j \in \mathcal{M}_i, \forall i, j \in \mathcal{M}$. We assume that each BS is equipped with an ES and subscribed by a set of UEs which randomly generate computational-intensive tasks that should be offloaded to the ESs due to their limited computation and energy resources. We use set $\mathcal{N}_i$ to denote the UEs associated with BS i and let $\mathcal{N}$ collect all UEs, i.e., $\mathcal{N} = \sum_{i\in\mathcal{M}} \mathcal{N}_i$. The total number of UEs is $N = |\mathcal{N}|$. In the considered SC-MEC network, we allow the UEs and their tasks to range from different types.

B. MmWave Channel and Backhaul Rate Models

To combat the significant propagation loss of mmWave channels, directional antennas and beamforming technologies are widely adopted. For analytical tractability, we apply a commonly used sectored antenna pattern model [9], which captures the fundamental characteristics of antenna patterns, such as directive gains, front-to-back ratio, half-power beamwidth, and so on. Specifically, the gains in the sector antenna pattern are a constant for all angles within the main lobe and equal to a smaller constant in the side lobes. Let $\theta_{ij}^{\mathrm{Tx}}(t)$ and $\theta_{ij}^{\mathrm{Rx}}(t)$ denote the beam-level beamwidth at transmitter i and receiver j at slot t, respectively. Let $\omega_{ij}^{\mathrm{Tx}}$ and $\omega_{ij}^{\mathrm{Rx}}$ be the angles deviating from the strongest path between the transmitter i and receiver j, respectively. The transmission gain $G_{ij}^{\mathrm{Tx}}(\omega_{ij}^{\mathrm{Tx}}, \theta_{ij}^{\mathrm{Tx}}(t))$ at transmitter i and the reception gain $G_{ij}^{\mathrm{Rx}}(\omega_{ij}^{\mathrm{Rx}}, \theta_{ij}^{\mathrm{Rx}}(t))$ at receiver j can be calculated by

$$G_{ij}^{\mathrm{Tx}}(\omega_{ij}^{\mathrm{Tx}}, \theta_{ij}^{\mathrm{Tx}}(t)) = \begin{cases} \dfrac{2\pi - (2\pi - \theta_{ij}^{\mathrm{Tx}}(t))\Gamma}{\theta_{ij}^{\mathrm{Tx}}(t)}, & \text{if } |\omega_{ij}^{\mathrm{Tx}}| \le \dfrac{\theta_{ij}^{\mathrm{Tx}}(t)}{2}, \\ \Gamma, & \text{otherwise}, \end{cases} \tag{1}$$

$$G_{ij}^{\mathrm{Rx}}(\omega_{ij}^{\mathrm{Rx}}, \theta_{ij}^{\mathrm{Rx}}(t)) = \begin{cases} \dfrac{2\pi - (2\pi - \theta_{ij}^{\mathrm{Rx}}(t))\Gamma}{\theta_{ij}^{\mathrm{Rx}}(t)}, & \text{if } |\omega_{ij}^{\mathrm{Rx}}| \le \dfrac{\theta_{ij}^{\mathrm{Rx}}(t)}{2}, \\ \Gamma, & \text{otherwise}, \end{cases} \tag{2}$$

where $0 < \Gamma \ll 1$ is the side lobe gain. In practice, $\Gamma \ll 1$ for narrow beams, i.e., $\theta_{ij}^{\mathrm{Tx}}(t)$ and $\theta_{ij}^{\mathrm{Rx}}(t)$ are small.

Combing the spatial channel power gain $G_{ij}(t)$ which involves the path loss, shadowing and small-scale fading, the overall effective channel power gain of backhaul link (i, j) is

$$h_{ij}(t) = G_{ij}^{\mathrm{Tx}}(\omega_{ij}^{\mathrm{Tx}}, \theta_{ij}^{\mathrm{Tx}}(t)) G_{ij}(t) G_{ij}^{\mathrm{Rx}}(\omega_{ij}^{\mathrm{Rx}}, \theta_{ij}^{\mathrm{Rx}}(t)), \forall t. \tag{3}$$

We consider a link-level transmit power allocation in this paper. Let $p_{ij}(t)$ be the transmit power allocated by BS i to BS j over mmWave backhaul link (i, j) at slot t. We have

$$\sum_{j\in\mathcal{M}_i} p_{ij}(t) \le P_i^{\max}, \forall i \in \mathcal{M}, t, \tag{4a}$$

$$p_{ij}(t) \ge 0, \forall i \in \mathcal{M}, j \in \mathcal{M}_i, t, \tag{4b}$$

where $P_i^{\max}$ is the maximum allowed transmit power of BS i. As a result, the Shannon's capacity of mmWave backhaul link (i, j) can be computed by

$$\mu_{ij}(t) = W \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{W N_j^0 + I_{ij}(t)} \right), \forall t, \tag{5}$$

where W is the system spectrum bandwidth, N_j^0 is the power spectral density of the additive white Gaussian noise at BS j, and $I_{ij}(t) = \sum_{i'\neq i} \sum_{j'\in\mathcal{M}_{i'}} p_{i'j'}(t) h_{i'j}(t)$ is the overall interference power for signals of BS i at BS j.

C. Task Admission and Queue Models

We consider delay-tolerant and data-oriented computation tasks [1], [6], which can be divided and buffered in task queues for distributed and parallel computing. The tasks of each UE n received by its associated BS at slot t is characterized by a triplet $(A_n(t), \zeta_n A_n(t), \varsigma_n A_n(t))$, where $A_n(t)$ is the task size (in bits), ζ_n is the required CPU-cycles for processing one task bit, and ς_n is the result output per task bit. Due to the limited network resources, only a part of tasks can be admitted for each UE. We denote the admitted UE n's tasks by $a_n(t)$, satisfying

$$0 \le a_n(t) \le A_n(t) \le A_n^{\max}, \forall t, \tag{6}$$

where $A_n^{\max}$ is the maximum size of $A_n(t)$ for all slots.

Instead of investigating the one-hop UE-BS task offloading, we study the multi-hop BS-BS cooperative task offloading to improve the utilization of geographical communication and computation resources. Specifically, the UE tasks can be not only computed by the local ESs but also offloaded to other ESs for cooperative computing through multi-hop mmWave backhauls. To this end, each BS needs to maintain N task queues buffering unprocessed tasks and N result queues buffering the task computation results. Let $Q_i^n(t)$ and $R_i^n(t)$ be UE n's task queue backlog and result queue backlog at BS i at slot t. The queue dynamics are given by

$$\begin{aligned} Q_i^n(t+1) = \max\{&Q_i^n(t) - \tau \sum_{j\in\mathcal{M}_i} \mu_{ij}^{n,\mathrm{Q}}(t) - f_i^n(t)/\zeta_n, 0\} \\ &+ \tau \sum_{j\in\mathcal{M}_i} \mu_{ji}^{n,\mathrm{Q}}(t) + \mathbf{1}\{n\in\mathcal{N}_i\}a_n(t), \end{aligned} \tag{7a}$$

$$\begin{aligned} R_i^n(t+1) = \max\{&R_i^n(t) - \tau \sum_{j\in\mathcal{M}_i} \mu_{ij}^{n,\mathrm{R}}(t), 0\} \\ &+ \tau \sum_{j\in\mathcal{M}_i} \mu_{ji}^{n,\mathrm{R}}(t) + \varsigma_n f_i^n(t)/\zeta_n, \end{aligned} \tag{7b}$$

where $\mathbf{1}\{\cdot\}$ is an indicator function which takes value 1 if the term in the brace holds true, and 0 otherwise. We let $R_i^n(t) = 0, \forall t$, if $n \in \mathcal{N}_i$. $\mu_{ij}^{n,\mathrm{Q}}(t)$ and $\mu_{ij}^{n,\mathrm{R}}(t)$ are the transmit rates allocated by BS i to UE n over backhual link (i,j) for transmitting the unprocessed tasks and task results. The transmit rate allocation for BS i should satisfy

$$\sum_{n\in\mathcal{N}} \mu_{ij}^{n,\mathrm{Q}}(t) + \mu_{ij}^{n,\mathrm{R}}(t) \le \mu_{ij}(t), \forall j \in \mathcal{M}_i, t, \tag{8a}$$

$$\mu_{ij}^{n,\mathrm{Q}}(t), \mu_{ij}^{n,\mathrm{R}}(t) \ge 0, \forall j \in \mathcal{M}_i, n \in \mathcal{N}, t. \tag{8b}$$

$f_i^n(t)$ in (7) represents the CPU-cycles allocated by the ES at BS i to UE n at slot t, while $f_i^n(t)/\zeta_n$ and $\varsigma_n f_i^n(t)/\zeta_n$ are the amounts of the computed tasks and result output of UE n, respectively. The CPU-cycle allocation should satisfy

$$\sum_{n\in\mathcal{N}} f_i^n(t) \le F_i^{\max}, \forall i \in \mathcal{M}, t, \tag{9a}$$

$$f_i^n(t) \ge 0, \forall i \in \mathcal{M}, n \in \mathcal{N}, t, \tag{9b}$$

where $F_i^{\max}$ is the computational capability of the ES at BS i.

According to the Little's theorem, the average delay is proportional to the average queue backlog, for a given task arrival rate. Therefore, we introduce the concept of queue strong stability [1] to bound the time-averaged queue backlog.

Definition 1. *The SC-MEC network is strongly stable if the sum total of the task queues and result queues is strongly stable, i.e.,*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i\in\mathcal{M}} \sum_{n\in\mathcal{N}} \mathbb{E}[Q_i^n(t) + R_i^n(t)] < \infty. \tag{10}$$

III. Problem Formulation

In this section, we formulate the EE maximization problem. The energy consumption of BS i is calculated by $e_i^{\mathrm{BS}}(t) = \tau\varphi_i \sum_{j\in\mathcal{M}_i} p_{ij}(t) + \tau P_i^{\mathrm{static}}$, and that of the ES at BS i is calculated by $e_i^{\mathrm{ES}}(t) = \psi_i \sum_{n\in\mathcal{N}} f_i^n(t) + \tau F_i^{\mathrm{static}}$, where φ_i stands for the power amplifier efficiency factor of the BS transmitter, ψ_i denotes the energy consumption per CPU-cycle of the ES, P_i^{static} denotes the static power consumption of the BS, and F_i^{static} is the static power consumption of the ES. Accordingly, the total energy consumption at slot t is

$$e(t) = \sum_{i\in\mathcal{M}} e_i^{\mathrm{BS}}(t) + e_i^{\mathrm{ES}}(t). \tag{11}$$

For each UE n, we define a logarithmic utility function $U_n(t) = \rho\log(1 + a_n(t)/\rho)$ where $\rho > 0$ is any real number. Note that $U_n(t)$ is a non-decreasing, differentiable and concave function over the interval $0 \le a_n(t) \le A_n^{\max}$, which is often used to reflect the well-known proportional fairness. The overall network utility can be therefore given by

$$U(t) = \sum_{n\in\mathcal{N}} \rho\log(1 + a_n(t)/\rho). \tag{12}$$

Let $\boldsymbol{\omega}(t) = \{A_n(t), h_{ij}(t), \forall n \in \mathcal{N}, i, j \in \mathcal{M}\}$ be the instantaneous network random state at slot t. Let $\boldsymbol{\alpha}(t) = \{a_n(t), p_{ij}(t), \mu_{ij}^{n,\mathrm{Q}}(t), \mu_{ij}^{n,\mathrm{R}}(t), f_i^n(t), \forall n \in \mathcal{N}, i, j \in \mathcal{M}\}$ be the decision action chosen in reaction to the network state $\boldsymbol{\omega}(t)$. The energy consumption $e(t)$ and the network utility $U(t)$ are affected by $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\omega}(t)$, and therefore can be written as $e(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$ and $U(\boldsymbol{\alpha}(t), \boldsymbol{\omega}(t))$, respectively. We now define the EE by $\overline{\eta} = \overline{U(\boldsymbol{\alpha}, \boldsymbol{\omega})}/\overline{e(\boldsymbol{\alpha}, \boldsymbol{\omega})}$ where $\overline{X} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[X(t)]$ for a given sequence $\{X(0), X(1), \ldots\}$. The problem of interest is formulated as

$$\overline{\eta}^* = \max_{\boldsymbol{\alpha}} \frac{\overline{U(\boldsymbol{\alpha}, \boldsymbol{\omega})}}{\overline{e(\boldsymbol{\alpha}, \boldsymbol{\omega})}} \tag{13a}$$

$$\text{s.t. } (4) - (10), \tag{13b}$$

$$\overline{a}_n \ge a_n^{\mathrm{req}}, \forall n \in \mathcal{N}, \tag{13c}$$

where $\overline{\eta}^*$ is the optimal EE. Constraint (13c) represents the time-averaged task admission rate requirement for each UE. Note that problem (13) is vexed due to the unknown distribution of $\boldsymbol{\omega}$, the non-convex fractional objective function, as well as the rate requirement (13c). To handle the fractional objective, we exploit the following Dinkelbach's theorem [8] in fractional programming theory to transform the problem into a tractable subtractive form. In particular, let $\boldsymbol{\alpha}^*$ be an optimal policy, then we have:

Theorem 1. *(Dinkelbach's Theorem [8]) The optimal EE $\overline{\eta}^*$ can be achieved by the optimal policy $\boldsymbol{\alpha}^*$ if and only if*

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \overline{U(\boldsymbol{\alpha}, \boldsymbol{\omega})} - \overline{\eta}^* \overline{e(\boldsymbol{\alpha}, \boldsymbol{\omega})} \\ &\quad = \overline{U(\boldsymbol{\alpha}^*, \boldsymbol{\omega})} - \overline{\eta}^* \overline{e(\boldsymbol{\alpha}^*, \boldsymbol{\omega})} = 0. \end{aligned} \tag{14}$$

A similar proof of Theorem 1 can be found in [7]. We obviously know through Theorem 1 that problem (13) is equivalent to the following problem upon the given optimal

$\overline{\eta}^*$, i.e.,

$$\begin{aligned}&\max_{\boldsymbol{\alpha}} \overline{U(\boldsymbol{\alpha},\boldsymbol{\omega})} - \overline{\eta}^* \overline{e(\boldsymbol{\alpha},\boldsymbol{\omega})}\\ &\text{s.t. } (4) - (10), (13\text{c}).\end{aligned} \tag{15}$$

Problem (15) is more tractable than its original version. We then utilize the virtual queue technology [1] to convert constraint (13c). Let $Y_n(t)$ be the virtual queue assigned to UE n by its associated BS. $Y_n(t)$ is initially empty and updated by

$$Y_n(t+1) = \max\{Y_n(t) - a_n(t), 0\} + a_n^{\text{req}}, \tag{16}$$

where a_n^{req} performs as the input and $a_n(t)$ performs as the output of the virtual queue $Y_n(t)$. To stabilize $Y_n(t)$, the time-averaged output $\overline{a}_n$ should not be less than the time-averaged input $\overline{a}_n^{\text{req}} = a_n^{\text{req}}$, i.e., $\overline{a}_n \geq a_n^{\text{req}}$. Therefore, we can replace the task admission rate requirement (13c) with the strong stability of the virtual queue, i.e.,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Y_n(t)] < \infty, \forall n \in \mathcal{N}, \tag{17}$$

Consequently, the problem can be further reformulated as

$$\begin{aligned}&\max_{\boldsymbol{\alpha}} \overline{U(\boldsymbol{\alpha},\boldsymbol{\omega})} - \overline{\eta}^* \overline{e(\boldsymbol{\alpha},\boldsymbol{\omega})}\\ &\text{s.t. } (4) - (10), (17).\end{aligned} \tag{18}$$

So far, we have transformed the original problem (13) to problem (18). In the next section, we will propose a Lyapunov optimization technique based adaptive policy without requiring the distribution information of $\boldsymbol{\omega}$.

IV. Adaptive Cooperative Task Offloading

A. Lyapunov Drift-Plus-Penalty

Let $\boldsymbol{\Theta}(t) = \{Q_i^n(t), R_i^n(t), Y_n(t), \forall i \in \mathcal{M}, n \in \mathcal{N}\}$ collects backlogs of all queues. Define the following quadratic Lyapunov function to measure the queue backlog level

$$L(\boldsymbol{\Theta}(t)) = \frac{1}{2}\sum_{i\in\mathcal{M}}\sum_{n\in\mathcal{N}} (Q_i^n(t)^2 + R_i^n(t)^2) + \frac{1}{2}\sum_{n\in\mathcal{N}} Y_n(t)^2, \tag{19}$$

Then the one-slot conditional Lyapunov drift is written as

$$\Delta(\boldsymbol{\Theta}(t)) = \mathbb{E}[L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) | \boldsymbol{\Theta}(t)]. \tag{20}$$

By adding the objective penalty term $-\mathbb{E}[U(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) - \overline{\eta}^* e(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) | \boldsymbol{\Theta}(t)]$ weighting a control parameter V to (20), we obtain the Lyapunov drift-plus-penalty as follows

$$\Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}[U(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) - \overline{\eta}^* e(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) | \boldsymbol{\Theta}(t)]. \tag{21}$$

The basic idea of Lyapunov optimization is to minimize the Lyapunov drift-plus-penalty, so that the objective value can be improved and meanwhile the queue backlog can be pushed into a lower level. However, directly minimizing (21) is still difficult. We thus pursue minimizing an upper bound of the Lyapunov drift-plus-penalty, which is given by Lemma 1.

Lemma 1. *For any feasible decision policy $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}(t), \forall t\}$, possible value of $\boldsymbol{\Theta}(t), \forall t$, and control parameter $V > 0$, the Lyapunov drift-plus-penalty is upper bounded by*

$$\begin{aligned}&\Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}[U(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) - \overline{\eta}^* e(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) | \boldsymbol{\Theta}(t)]\\ &\leq B - V\mathbb{E}[U(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) - \overline{\eta}^* e(\boldsymbol{\alpha}(t),\boldsymbol{\omega}(t)) | \boldsymbol{\Theta}(t)]\\ &\quad + \sum_{i\in\mathcal{M}}\sum_{n\in\mathcal{N}} Q_i^n(t)\mathbb{E}\Bigg[\tau \sum_{j\in\mathcal{M}_i} \left(\mu_{ji}^{n,\text{Q}}(t) - \mu_{ij}^{n,\text{Q}}(t)\right)\\ &\qquad + \mathbf{1}\{n\in\mathcal{N}_i\} a_n(t) - f_i^n(t)/\zeta_n \Bigg| \boldsymbol{\Theta}(t)\Bigg]\\ &\quad + \sum_{i\in\mathcal{M}}\sum_{n\in\mathcal{N}} R_i^n(t)\mathbb{E}\Bigg[\tau \sum_{j\in\mathcal{M}_i} \left(\mu_{ji}^{n,\text{R}}(t) - \mu_{ij}^{n,\text{R}}(t)\right)\\ &\qquad + \varsigma_n f_i^n(t)/\zeta_n \Bigg| \boldsymbol{\Theta}(t)\Bigg]\\ &\quad + \sum_{n\in\mathcal{N}} Y_n(t)\mathbb{E}[a_n^{\text{req}} - a_n(t) | \boldsymbol{\Theta}(t)],\end{aligned} \tag{22}$$

where B is a constant related to the maximum secondary moments of the decision variables.

Proof: We omit the proof due to the limited space.

B. Algorithm Design

We propose the ACTO algorithm by opportunistically minimizing the right hand side (RHS) of (22) at each slot t based on the instantaneous observations of network state $\boldsymbol{\omega}(t)$ and queue backlog $\boldsymbol{\Theta}(t)$. However, the optimal $\overline{\eta}^*$ cannot be easily obtained in advance. We thus utilize an estimate of $\overline{\eta}^*$ by taking advantage of its running average, which is given by

$$\overline{\eta}(t) = \frac{\frac{1}{t}\sum_{t_0=0}^{t-1} U(\boldsymbol{\alpha}(t_0),\boldsymbol{\omega}(t_0))}{\frac{1}{t}\sum_{t_0=0}^{t-1} e(\boldsymbol{\alpha}(t_0),\boldsymbol{\omega}(t_0))}. \tag{23}$$

By decoupling from the RHS of (22), the ACTO algorithm solve the following subproblems at each slot.

1) Task Admission: At each slot t, the amount of tasks admitted for UE n by its associated BS i^n (i.e., $n \in \mathcal{N}_{i^n}$) can be determined by solving

$$\begin{aligned}&\max_{a_n} V\rho\log(1 + a_n/\rho) - (Y_n(t) - Q_{i^n}^n(t))a_n\\ &\text{s.t. } 0 \leq a_n \leq A_n(t).\end{aligned} \tag{24}$$

Problem (24) is an one-dimensional convex programming whose optimal solution can be achieved by

$$a_n(t) = \begin{cases} 0, & \text{if } Q_{i^n}^n(t) - Y_n(t) \geq V,\\ A_n(t), & \text{if } Q_{i^n}^n(t) - Y_n(t) \leq 0,\\ \min\{\dfrac{V\rho}{Q_{i^n}^n(t) - Y_n(t)} - \rho, A_n(t)\}, & \text{otherwise.}\end{cases} \tag{25}$$

2) Joint Transmit Rate and Power Allocation: At each slot t, the joint transmit rate and power allocation subproblem for each BS i can be given by

$$\begin{aligned}\max_{\{\mu_{ij}^{n,\text{Q}},\mu_{ij}^{n,\text{R}},p_{ij},\forall j\in\mathcal{M}_i,n\in\mathcal{N}\}} &\Bigg\{ \sum_{j\in\mathcal{M}_i}\sum_{n\in\mathcal{N}} \left[\left(Q_i^n(t) - Q_j^n(t)\right)\mu_{ij}^{n,\text{Q}}\right.\\ &\left. + \left(R_i^n(t) - R_j^n(t)\right)\mu_{ij}^{n,\text{R}}\right] - V\overline{\eta}(t)\sum_{j\in\mathcal{M}_i} \varphi_i p_{ij}\Bigg\}\end{aligned}$$

$$\begin{aligned}
\text{s.t.} & \sum_{n\in\mathcal{N}} \mu_{ij}^{n,\mathrm{Q}} + \mu_{ij}^{n,\mathrm{R}} \le \mu_{ij}(t), \forall j \in \mathcal{M}_i, \\
& \sum_{j\in\mathcal{M}_i} p_{ij} \le P_i^{\max}, p_{ij} \ge 0, j \in \mathcal{M}_i, \\
& \mu_{ij}^{n,\mathrm{Q}}, \mu_{ij}^{n,\mathrm{R}} \ge 0, \forall j \in \mathcal{M}_i, n \in \mathcal{N}. \qquad (26)
\end{aligned}$$

Note that the mmWave backhaul capacity $\mu_{ij}(t)$ is a function of transmit powers according to (5). Therefore, for decoupling we first solve the transmit rate allocation given the optimal transmit power $\mathbf{p}^*(t) = \{p_{ij}^*(t), \forall i, j \in \mathcal{M}\}$. By substituting $\mu_{ij}(t) = \mu_{ij}(\mathbf{p}^*(t))$ and $p_{ij} = p_{ij}^*(t), \forall j \in \mathcal{M}_i$ into (26), the transmit rate allocation reduces to a linear programming problem of weighted-sum maximization, which can be solved by Algorithm 1. $\Upsilon_{ij}^{\mathrm{Q}}(t)$ and $\Upsilon_{ij}^{\mathrm{R}}(t)$ in Algorithm 1 are the largest weights of the first term and the second term for link (i, j) in the objective of (26), respectively, i.e.,

$$\Upsilon_{ij}^{\mathrm{Q}}(t) = \max\{Q_i^n(t) - Q_j^n(t), \forall n \in \mathcal{N}\}, \qquad (27a)$$

$$\Upsilon_{ij}^{\mathrm{R}}(t) = \max\{R_i^n(t) - R_j^n(t), \forall n \in \mathcal{N}\}. \qquad (27b)$$

According to Algorithm 1, it is obvious that no transmit rate would be allocated to UEs over each backhaul link (i, j) if $\Upsilon_{ij}(t) = \max\{\Upsilon_{ij}^{\mathrm{Q}}(t), \Upsilon_{ij}^{\mathrm{R}}(t)\} \le 0$, in which case, no transmit power is required, and therefore we can simply set $p_{ij}(t) = 0$. For each BS i, we can define a subset $\mathcal{M}_i^+ = \{j | \Upsilon_{ij}(t) > 0, \forall j \in \mathcal{M}_i\}$. Then, the derived transmit power allocation problem can be expressed as

$$\begin{aligned}
\max_{\{p_{ij}, \forall j \in \mathcal{M}_i^+\}} & \left\{ \sum_{j\in\mathcal{M}_i^+} \Upsilon_{ij}(t) W \log_2\left(1 + \frac{p_{ij} h_{ij}(t)}{W N_j^0 + \overline{I}_{ij}(t)}\right) \right. \\
& \left. - V\overline{\eta}(t)\varphi_i p_{ij} \right\} \\
\text{s.t.} & \sum_{j\in\mathcal{M}_i^+} p_{ij} \le P_i^{\max}, \\
& p_{ij} \ge 0, j \in \mathcal{M}_i^+, \qquad (28)
\end{aligned}$$

where $\overline{I}_{ij}(t) = \frac{1}{t}\sum_{t_0=0}^{t-1} I_{ij}(t_0)$ is the running-average approximation of the real $I_{ij}(t)$. This can be done by using some interference measure technologies [10]. Problem (28) is a convex programming which can be solved by using the bisection search based water filling method [1].

3) CPU-Cycle Allocation: At each slot t, each ES at BS i allocates CPU-cycles to UEs by solving

$$\begin{aligned}
\max_{\{f_i^n, \forall n \in \mathcal{N}\}} & \sum_{n\in\mathcal{N}} [(Q_i^n(t) - \varsigma_n R_i^n(t))/\zeta_n - V\overline{\eta}(t)\psi_i] f_i^n \\
\text{s.t.} & \sum_{n\in\mathcal{N}} f_i^n \le F_i^{\max}, \qquad (29) \\
& f_i^n \ge 0, n \in \mathcal{N}.
\end{aligned}$$

Problem (29) is a linear programming problem of weighted sum maximization whose optimal solution can be found by

$$f_i^n(t) = \begin{cases} F_i^{\max}, & \text{if } n = \underset{k\in\mathcal{N}}{\operatorname{argmax}}\, \Xi_i^k(t) \text{ and } \Xi_i^n(t) > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad (30)$$

Algorithm 1 Transmit Rate Allocation Algorithm for BS i

```
Input: $\Upsilon_{ij}^{\mathrm{Q}}(t), \Upsilon_{ij}^{\mathrm{R}}(t), \mu_{ij}(\mathbf{p}^*(t)), \forall j \in \mathcal{M}_i$;
Output: $\mu_{ij}^{n,\mathrm{Q}}(t), \mu_{ij}^{n,\mathrm{R}}(t), \forall j \in \mathcal{M}_i$;
for all backhaul link $(i, j), \forall j \in \mathcal{M}_i$ do
    if $\Upsilon_{ij}^{\mathrm{Q}}(t) \le \Upsilon_{ij}^{\mathrm{R}}(t)$ then
        Set $\mu_{ij}^{n,\mathrm{Q}}(t) = 0, \forall n \in \mathcal{N}$;
        if $\Upsilon_{ij}^{\mathrm{R}}(t) \le 0$ then
            Set $\mu_{ij}^{n,\mathrm{R}}(t) = 0, \forall n \in \mathcal{N}$;
        else
            Find $n' = \mathrm{argmax}_{k\in\mathcal{N}}\{R_i^k(t) - R_j^k(t)$;
            Set $\mu_{ij}^{n',\mathrm{R}}(t) = \mu_{ij}(\mathbf{p}^*(t))$;
            Set $\mu_{ij}^{n,\mathrm{R}}(t) = 0, \forall n \in \mathcal{N} \setminus \{n'\}$;
        end if
    else
        Set $\mu_{ij}^{n,\mathrm{R}}(t) = 0, \forall n \in \mathcal{N}$;
        if $\Upsilon_{ij}^{\mathrm{Q}}(t) \le 0$ then
            Set $\mu_{ij}^{n,\mathrm{Q}}(t) = 0, \forall n \in \mathcal{N}$;
        else
            Find $n' = \mathrm{argmax}_{k\in\mathcal{N}}\{Q_i^k(t) - Q_j^k(t)$;
            Set $\mu_{ij}^{n',\mathrm{Q}}(t) = \mu_{ij}(\mathbf{p}^*(t))$;
            Set $\mu_{ij}^{n,\mathrm{Q}}(t) = 0, \forall n \in \mathcal{N} \setminus \{n'\}$;
        end if
    end if
end for
```

where $\Xi_i^n(t) = (Q_i^n(t) - \varsigma_n R_i^n(t))/\zeta_n - V\overline{\eta}(t)\psi_i, \forall n \in \mathcal{N}$.

Generally, it can be proved that the proposed ACTO algorithm can asymptotically achieve an $O(V)$ time-averaged queue backlog and $O(1/V)$ optimality of the EE. We will elaborate the performance results in our future work.

V. Simulation Results

We evaluate the ACTO algorithm by simulating one SC-MEC network with one macro BS and 4 small BSs for $T = 1.2 \times 10^5$ time slots. The mmWave backhaul channel gain is simulated based on a commonly used sectored antenna model in [11]. The rest of main simulation parameters are listed in Table I. We adopt two baselines for algorithm comparison: (a) Static Cooperative Task Offloading (SCTO), i.e., the transmit power of each BS and the CPU-cycles of each ES are equally allocated to its connected BSs and all UEs, respectively, while the rest of algorithm steps are as the same as those of ACTO; (b) Adaptive Non-Cooperative Task Offloading (ANCTO), i.e., the UE tasks are computed locally by their associated ESs with adaptive task admission and CPU-cycle allocation.

Fig. 1 shows that the time-averaged EE $\overline{\eta}(T)$ increases as the number of UEs grows under both ACTO and SCTO algorithms. This is because the growth of UEs provides more degrees of freedom for the problem optimization, which enhances the network utility with a relatively small amount of extra energy consumption. It can be observed that our ACTO algorithm achieves better EE performance than the SCTO algorithm. Another insight from Fig. 1 is that the EE value under a larger V (i.e., 4×10^5) is higher than that under a

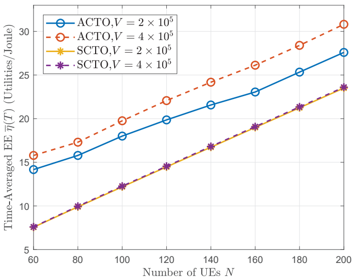

Fig. 1. Time-Averaged EE versus UE Number

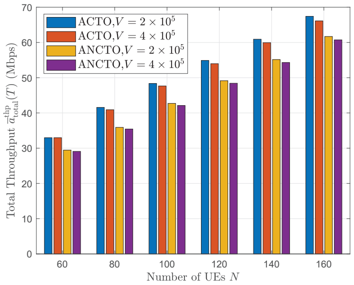

Fig. 2. Total Throughput versus UE Number

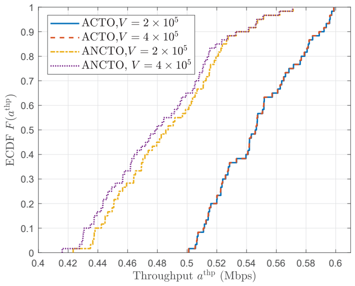

Fig. 3. ECDF of UE Throughput

TABLE I
SIMULATION PARAMETERS

Parameters	Values
W	20 MHz
$WN_j^0, \forall j \in \mathcal{M}$	-174 dBm/Hz $+10\log_{10} W$
$P_0^{\text{max}}, P_0^{\text{static}}, F_0^{\text{max}}$	40 Watt, 63 Watt, 200 Watt
$P_i^{\text{max}}, P_i^{\text{static}}, F_i^{\text{max}}, \forall i \in \mathcal{M} \setminus \{0\}$	1 Watt, 4 Watt, 20 Watt
$\varphi_i, \forall i \in \mathcal{M}$	2
$F_0^{\text{max}}; F_i^{\text{max}}, \forall i \in \mathcal{M} \setminus \{0\}$	30 Megacycles; 10 Megacycles
$\psi_0; \psi_i, \forall i \in \mathcal{M} \setminus \{0\}$	9×10^{-6} Joule/CPU-cycle; 1×10^{-6} Joule/CPU-cycle
$A_n^{\text{max}}, a_n^{\text{req}}, \forall n \in \mathcal{N}$	$[4, 6]$ Kbits, $[0.5, 0.6]$ Kbits
$\zeta_n, \forall n \in \mathcal{N}$	$[500, 1500]$ CPU-cycles/bit
$\varsigma_n, \forall n \in \mathcal{N}$	$[0.1, 1]$
ρ	10
τ	1 ms

smaller V (i.e., 2×10^5). In other words, the EE optimality gap would diminish gradually with V increasing, which therefore verifies the statement given at the end of Section IV.

Fig. 2 depicts the total throughput which is defined by $\overline{a}^{\text{thp}}_{\text{total}}(T) = \frac{1}{T\tau} \sum_{t=0}^{T-1} \sum_{n \in \mathcal{N}} a_n(t)$, versus the number of UEs N under both ACTO and ANCTO. We can see that increasing N helps with the growth of $\overline{a}^{\text{thp}}_{\text{total}}(T)$ under both algorithms. For a given N and V, the proposed ACTO algorithm outperforms the ANCTO algorithm in terms of the total throughput, which demonstrates the benefit of the BS/ES cooperation.

Fig. 3 shows the empirical cumulative distribution function (ECDF) of the UE throughput with $N = 60$ under ACTO and ANCTO. The ECDF is defined by $F(a^{\text{thp}}) = \frac{\sum_{n \in \mathcal{N}} \mathbf{1}\{\overline{a}_n^{\text{thp}}(T) \le a^{\text{thp}}\}}{N}$, where $\overline{a}_n^{\text{thp}}(T) = \frac{1}{T\tau} \sum_{t=0}^{T-1} a_n(t), \forall n \in \mathcal{N}$. It can be observed that the ECDF under ANCTO rises up to the 1 probability faster than that under ACTO as a^{thp} increases, which implies that ACTO yields higher individual UE throughput than ANCTO. Furthermore, we can see that under the ANCTO algorithm the throughputs of roughly 58% UEs under $V = 2 \times 10^5$ and 65% UEs under $V = 4 \times 10^5$ are less than 0.5 Mbps, while all UEs' throughputs belong to $[0.5, 0.6]$ Mbps for our ACTO under the same parameter settings. Since $a_n^{\text{req}}/\tau \in [0.5, 0.6]$ Mbps according to Table I, we can conclude that the proposed ACTO algorithm outperforms ANCTO in guaranteeing the UE throughput (task admission rate) requirements benefitting from the BS/ES cooperation.

VI. CONCLUSIONS

We investigated the adaptive cooperative task offloading for EE maximization in mmWave backhaul enabled SC-MEC networks in this paper. A stochastic EE maximization problem was formulated subject to the stability constraints of task and result queues, as well as the requirements of UE task admission rates. An ACTO algorithm which does not require a-priori knowledge of the network dynamics was proposed based on the Lyapunov optimization and fractional programming theory. The proposed algorithm outperforms the static cooperative algorithm in EE and the adaptive non-cooperative algorithm in throughput, respectively.

VII. ACKNOWLEDGEMENT

This work was supported in part by CSC Scholarship (CSC202006960079), in part by NSF China (61901319, 61971327), and in part by NRF of Korea-Grant funded by the Korean Government (MSIT)-NRF-2020R1A2B5B02002478.

REFERENCES

[1] Z. Jing, Q. Yang, M. Qin, J. Li, and K. S. Kwak, "Long-term maxmin fairness guarantee mechanism for integrated multi-RAT and MEC metworks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2478-2492, Mar. 2021.

[2] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651-2664, Dec. 2018.

[3] Y. Du, *et al.*, "Two-tier matching game in small cell networks for mobile edge computing," *IEEE Trans. Serv. Comput.*, Early Access.

[4] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031-5044, May 2019.

[5] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1619-1632, Aug. 2018.

[6] X. Lyu, *et al.*, "Distributed online optimization of fog computing for selfish devices with out-of-date information," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7704-7717, Nov. 2018.

[7] W. Dinkelbach, "On nonlinear fractional programming," *Manag. Sci.*, vol. 13, no. 7, pp. 492-498, Mar. 1967.

[8] T. K. Vu, M. Bennis, M. Debbah, and M. Latva-Aho, "Joint path selection and rate allocation framework for 5G self-backhauled mm-wave networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2431-2445, Apr. 2019.

[9] J. Wildman, P. H. J. Nardelli, M. Latva-aho, and S. Weber, "On the joint impact of beamwidth and orientation error on throughput in directional wireless poisson networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 7072-7085, Dec. 2014.

[10] S. Mishra, I. Wang, and S. N. Diggavi, "Harnessing bursty interference in multicarrier systems with output feedback," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4430-4452, Jul. 2017.

[11] M. R. Akdeniz, *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164-1179, Jun. 2014.