2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)

Joint D2D Collaboration and Task Offloading for Edge Computing: A Mean Field Graph Approach

Xiong Wang*, Jiancheng Ye†, and John C.S. Lui*
* Department of Computer Science and Engineering, The Chinese University of Hong Kong
† Network Technology Lab and Hong Kong Research Center, Huawei Technologies Co., Ltd.
E-mail: xwang@cse.cuhk.edu.hk, yejiancheng@huawei.com, johncslui@gmail.com

***Abstract*—Mobile edge computing (MEC) facilitates computation offloading to edge server, as well as task processing via device-to-device (D2D) collaboration. Existing works mainly focus on centralized network-assisted offloading solutions, which are unscalable to scenarios involving collaboration among massive users. In this paper, we propose a *joint framework* of decentralized D2D collaboration and efficient task offloading for a large-population MEC system. Specifically, we utilize the power of two choices for D2D collaboration, which enables users to beneficially assist each other in a decentralized manner. Due to short-range D2D communication and user movements, we formulate a *mean field model* on a finite-degree and dynamic graph to analyze the state evolution of D2D collaboration. We derive the existence, uniqueness and convergence of the state stationary point so as to provide a *tractable collaboration performance*. Complementing this D2D collaboration, we further build a *Stackelberg game* to model users' task offloading, where edge server is the leader to determine a service price, while users are followers to make offloading decisions. By embedding the Stackelberg game into Lyapunov optimization, we develop an *online* offloading and pricing scheme, which could optimize server's service utility and users' system cost simultaneously. Extensive evaluations show that our D2D collaboration can mitigate users' workloads by 73.8% and task offloading can achieve high energy efficiency.**

I. Introduction

In recent years, we have witnessed a rapid growth of data generated from the network edge, especially with the enormous popularity of mobile devices [1]. Many intelligent mobile applications, such as interactive gaming, real-time face recognition and natural language processing, are emerging which typically demand intensive computation and low latency. In general, mobile devices have constrained resources, while remote-resided cloud server suffers from high latency due to long-haul transmissions. To support the compute-intensive yet delay-sensitive applications, mobile edge computing (MEC) is recognized as a new paradigm to push cloud frontier close to the edge for such service requirements [2].

At a high level, MEC enables mobile users to offload tasks to the local edge server endowed with computing functionalities. Compared to the cloud datacenter, an individual edge server basically has limited computing capacity, making it difficult to accommodate huge amount of tasks since over 90% of the data will be stored and processed at the network edge [3]. Under this scenario, exploiting *collaboration among users* is a promising complementary approach to task offloading for easing the strain on the edge server [4]. Device-to-device (D2D) communication (e.g., via Bluetooth or Wi-Fi Direct) generally is more energy-saving while less time-consuming [9], thereby providing a low-latency service for users when they collaboratively process tasks via D2D links [5]. Specifically, heavily-loaded users can seek immediate assistance from lightly-loaded ones within proximity, and hence the average task delay is expected to decrease significantly.

978-0-7381-3207-5/21/$31.00 ©2021 IEEE

Despite the clear advantage of D2D collaboration, task offloading to edge server is still indispensable for MEC as high latency is witnessed if compute-intensive tasks are handled solely by resource-constrained mobile devices. Along with the offloading, a service price is charged by the edge server when providing computing service for mobile users [27]. Needless to say, setting a proper price is critical, as an excessively low price is insufficient to compensate for the server's operation cost, whereas an unduly high price will certainly cause a decrease in user demands of task offloading and further increase the delay in task execution. Therefore, a reasonable pricing scheme is required to incentivize task offloading while also bringing benefits to the edge server operator.

Various efforts have been dedicated to investigate D2D collaboration in MEC or fog computing [4]–[6]. Benefiting from users' mutual assistance, D2D collaboration can effectively improve energy efficiency and delay performance for MEC system. In these works, the edge server mainly serves as a central coordinator to aid the collaborative task processing, whereas the potential benefit of task offloading was not explored. Few following works further incorporate D2D collaboration into task offloading to take advantage of the computing capacity embedded in mobile devices and edge server [7], [8]. However, these works mostly concentrate on a centralized offloading and/or collaboration optimization, with a restrictive assumption of time-invariant D2D links in order to achieve a tractable analysis. When there are a large population of moving mobile users, which is often the case in D2D collaboration, *how to characterize a decentralized collaboration and develop an efficient offloading for "dynamic" MEC system* still remains unresolved. To answer this critical question, researchers are faced with the following challenges.

First, due to short-range D2D communication, collaboration occurs mainly among nearby users, thus leading to a graph structure formed by spatially distributed mobile users. A decentralized collaboration scheme should encompass both *static and dynamic connectivity setting* when considering user

978-1-6654-1494-4/21/$31.00 ©2021 IEEE

movements, which however is theoretically challenging in general. Therefore, it brings new modeling requirements for D2D collaboration to achieve rigorous theoretical guarantee as well as good empirical performance. Second, task offloading is influenced by the service price set by edge server, whereas the pricing scheme is also dependent on the strategic offloading decision of mobile users. Their *mutual-dependency* raises difficulty in the optimal offloading and pricing design. Third, D2D collaboration is *intertwined* with task offloading due to various task executions including collaborative execution and offloaded execution. This demands incorporating decentralized collaboration into determining appropriate proportions of tasks to be offloaded and to be processed locally so as to reduce the execution latency while enhancing energy efficiency.

In this paper, we propose a *joint D2D collaboration and task offloading* for a large-population MEC system. We first use the power of two (Po2) choices to enable a *decentralized collaboration* among massive mobile users, where each user randomly polls a neighbor within its D2D range and forwards a task if the polled neighbor has a lighter workload. We develop a novel *mean field model on graph* to analyze this D2D collaboration, through which we can characterize the state evolution of MEC system in both static and dynamic situations. By incorporating the steady state of D2D collaboration, we further formulate a Stackelberg game to model the task offloading from mobile users to edge server. Specifically, users are followers in making their offloading decisions, while edge server is the leader in determining a dynamic price based on *Lyapunov optimization* for providing computing service. As a result, we consider the intertwined collaboration and offloading processes so as to collectively promote an efficient task execution. This paper has the following main contributions:

- We develop a joint D2D collaboration and task offloading framework which facilitates users to collaboratively process tasks in a *decentralized manner* and offload computation to local edge server. Our framework targets the real-life large-population MEC system so as to fully unleash the potentials of widely-distributed mobile devices and the edge server's capacity. Evaluations show that we can reduce users' workloads by 73.8$\%$ and improve the energy efficiency as well.
- We propose a novel mean field model on static and dynamic graphs to characterize D2D collaboration, based on which we can analyze the stochastic state evolution by *deterministic ordinary differential equations* (ODEs). We rigorously prove the existence and uniqueness of mean field stationary point to provide a *theoretically tractable performance* for D2D collaboration. To the best of our knowledge, this is the first work that conducts a thorough analysis of mean field model on *finite-degree and dynamic graphs*.
- We design an *online offloading and pricing scheme* using a Lyapunov optimization framework to determine the optimal offloading and pricing decisions over time. By embedding a Stackelberg game into the online decision making, we can simultaneously minimize users' system cost while meeting their stringent task delay requirements, and maximize the server's *long-term utility* with only current information.

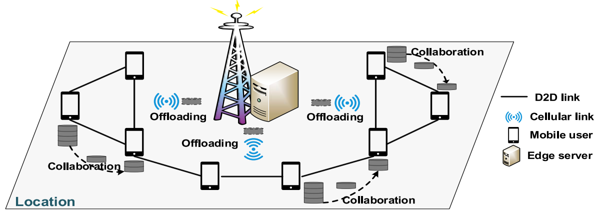

Fig. 1: Snapshot of the MEC system.

II. System Model

We consider a MEC system with large-population mobile users $\mathcal{N} = \{1, 2, ..., N\}$ and an edge server. Users' tasks can be offloaded to edge server via cellular network or processed by their collaboration via D2D link, as shown in Fig. 1.

A. System Overview

Due to short-range D2D communication, collaboration is mainly among users within proximity, and we model this collaboration structure as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with $\mathcal{E}$ denoting the D2D links. Besides, along with task offloading, a price p is charged by the edge server for providing computing service, where *p remains fixed* for a long period, e.g., weekly or monthly basis [26]. In return, users strategically choose to offload tasks with probability $x \in [0, 1]$, while handling the rest via D2D collaboration. A task is typically characterized by the required service time (CPU cycles) and the amount of cellular traffic (data size) [10]. By convention, task generation of each user follows a rate-λ Poisson process, where the service time of a task obeys an exponential distribution with normalized unit mean value, and the data size has an average value of B [11]. The normalized service rates of mobile devices and edge server are μ and γ, respectively, with $\lambda < \mu$ to keep the MEC system stable. Moreover, the server can process the offloaded tasks in parallel because it has a more powerful computing capability than mobile devices [12].

B. D2D Collaboration

We use the Po2 choices for decentralized collaboration [13]. Let $Q_u(t)$ be the number of tasks, or workload, of user $u \in \mathcal{N}$ at time t. For Po2, when a task is generated by u and not offloaded, u randomly polls a neighbor, say v, and forwards the task to v if $Q_u(t) > Q_v(t)$; otherwise the task joins $Q_u(t)$ with ties being broken arbitrarily. Denote d_u as the number of u's neighbors, also known as its degree in graph $\mathcal{G}$. The degree d_u is distributed in a *finite degree set* $\mathcal{K} = \{k_{\min}, ..., k_{\max}\}$ due to short-range D2D communication. W.l.o.g., the graph $\mathcal{G}$ is *uncorrelated* [14], i.e., the probability $p(k'|k)$ that a user with degree k has a link to a neighbor with degree k' satisfies:

$$p(k'|k) = \frac{k' p(k')}{\overline{k}}, \tag{1}$$

where $p(k)$ is the probability that a user has degree k and $\overline{k} = \mathbb{E}[k] = \sum_{k \in \mathcal{K}} p(k)k$ is the expected degree. Eq. (1) means the degree distribution of two neighboring users is independent.

For $k \in \mathcal{K}, i \geq 0$, we define $q_{k,i}(t)$ and $s_{k,i}(t)$ as:

$$q_{k,i}(t) = \frac{\sum_{u \in \mathcal{N}} \mathbf{1}_{d_u = k, Q_u(t) = i}}{\sum_{u \in \mathcal{N}} \mathbf{1}_{d_u = k}}, \tag{2}$$

$$s_{k,i}(t) = \sum\nolimits_{j \geq i} q_{k,j}(t). \tag{3}$$

Physically, $q_{k,i}(t)$ or $s_{k,i}(t)$ can be regarded as the probability that a user with degree k holds i or at least i tasks, respectively. Also, $q_{k,i}(t) = s_{k,i}(t) - s_{k,i+1}(t)$ and $s_{k,0}(t) = 1, \forall k \in \mathcal{K}$. We say $s_{k,1}(t)$ is the *busy probability* as it implies the case where a k-degree user has a non-empty workload. Denote $\boldsymbol{s}(t) = \{s_{k,i}(t)\}$ as the system state, that is, the workload distribution of the MEC system. Hereinafter, we will use $q_{k,i}, s_{k,i}, Q_u$ and $\boldsymbol{s}$ without index t if there is no confusion.

Our objective is to demonstrate that D2D collaboration can effectively mitigate users' workloads, by characterizing the stationary point $\boldsymbol{s}^*$ for state evolution when user number $N \to \infty$ and $\dot{\boldsymbol{s}}^* = \boldsymbol{0}$ (MEC system is stable). Given the graph collaboration structure, we develop a new analysis framework as existing Po2 methods are no longer applicable.

C. Task Offloading

Upon generating a task, users offload it to the edge server with probability x, and are charged with a price p. Since p is fixed for a long time, the offloading and pricing decisions are made in terms of discrete time slot $\{0, 1, ..., n, ...\}$, say at a time interval of every week. The probability $x[n]$ and price $p[n]$ are constant in each slot, and the MEC system is regarded to be stable. Hence, we can leverage the *stationary point* $\boldsymbol{s}^*$ in offloading and pricing scheme design.

1) Offloading constraint: Together with D2D collaboration, users also collaboratively decide the probability $x[n]$ in each time slot, which actually only relies on the information of $\boldsymbol{s}^*$.

Task delay. If a task is offloaded to the server, the expected task delay includes the transmission time and server processing time, that is $d_o = x[n](\frac{B}{r} + \frac{1}{\gamma})$ where r denotes the data rate of cellular network and 1 is the normalized unit service time. With probability $x_c[n] = 1 - x[n]$, a task will be processed via D2D collaboration, then the delay amounts to the queueing time, which is $d_q = x_c[n] \frac{\sum_i \sum_k p(k) s^*_{k,i}}{x_c[n]\lambda} = \frac{\sum_i \sum_k p(k) s^*_{k,i}}{\lambda}$ based on Little's law. Here, the transmission delay of fast and short-range D2D communication is negligibly small compared to d_o and d_q [9]. Therefore, the average task delay $d(x[n])$ is:

$$d(x[n]) = d_o + d_q. \tag{4}$$

Note that $s^*_{k,i}$ depends on $x_c[n]$, or offloading probability $x[n]$.

Collaboration fairness. Due to heterogeneous number of neighbors, users have *unbalanced contributions* in D2D collaboration, i.e., busy probability $s^*_{k,1}$ varies over degree k. When deciding the probability $x[n]$, collaboration fairness requires that the gap between the highest and lowest $s^*_{k,1}$ should not be too large so as to prevent the "free-riding" scenario.

2) Pricing constraint: The offloading probability $x[n]$ of mobile users is affected by the service price $p[n]$ charged by the edge server. In general, setting a high price will restrain user demands of task offloading, or low $x[n]$, whereas positing a low price will lead to *overloaded situation* at the edge server because of too many offloaded tasks. We consider that the server can adaptively choose $p[n]$ for compensating its operation cost and avoiding being overloaded.

3) Problem formulation: We now formulate the system cost of users and the service utility of the server in task offloading.

Users' system cost. Since D2D communication is energy efficient, the system cost of a user is mainly composed of charged fee, processing cost, and offloading transmission cost. Formally, the charged fee is the payment to the edge server for task offloading, which is $x[n]\lambda p[n]$. When processing a task, it needs an average $\frac{1}{\mu}$ time, so the energy consumption is $\rho_c^m \frac{1}{\mu}$ where ρ_c^m is the energy cost per CPU cycle for computation in a mobile device [10]. Moreover, the expected busy probability is $s_1^* \triangleq \sum_{k \in \mathcal{K}} p(k) s^*_{k,1}$ from Eq. (3). Therefore, the processing cost becomes $s_1^* \frac{\rho_c^m}{\mu}$. Finally, the transmission cost is $x[n]\lambda\rho_t^m \frac{B}{r}$ where ρ_t^m is the unit cost for transmitting cellular traffic. Overall, the system cost $c[n]$ in time slot n is:

$$c[n] = x[n]\lambda p[n] + s_1^* \frac{\rho_c^m}{\mu} + x[n]\lambda\rho_t^m \frac{B}{r}. \tag{5}$$

Server's service utility. On the server's side, its average profit in time slot n is $x[n]\lambda p[n]$, and average processing cost is $x[n]\lambda\frac{\rho_c^s}{\gamma}$ with ρ_c^s representing the energy cost per CPU cycle in the edge server. Then, the service utility $u[n]$ is acquired:

$$u[n] = x[n]\lambda p[n] - x[n]\lambda\frac{\rho_c^s}{\gamma}. \tag{6}$$

Stackelberg game. Given the price $p[n]$, users aim to reduce their system cost by deciding the probability $x[n]$, subject to constraints of task delay and collaboration fairness:

$$\begin{aligned} \min\nolimits_{x[n]} \quad & c[n] && (7) \\ \text{s.t.} \quad & d(x[n]) \leq \overline{d} && (7a) \\ & \max\{s^*_{k,1}\} - \min\{s^*_{k,1}\} \leq \overline{s}. && (7b) \end{aligned}$$

As for the edge server, its objective is to optimize the long-term utility via dynamically setting the service price $p[n]$:

$$\begin{aligned} \max\nolimits_{p[n]} \quad & \lim_{T \to \infty} \frac{1}{T} \sum\nolimits_{n=0}^{T-1} \mathbb{E}[u[n]] && (8) \\ \text{s.t.} \quad & \lim_{T \to \infty} \frac{1}{T} \sum\nolimits_{n=0}^{T-1} \mathbb{E}[x[n]\lambda] \leq \overline{x} && (8a) \\ & p[n] \in (0, p_u], && (8b) \end{aligned}$$

where the inequality in Eq. (8a) is the overloaded constraint and p_u denotes the highest price users could accept.

In time slot n, the server first chooses a price $p[n]$, and then users react via the offloading decision $x[n]$, which is modeled as a Stackelberg game. Note that optimizing cost and utility is intertwined with D2D collaboration. Also, we have to design an offloading and pricing scheme to simultaneously maximize service utility online and minimize system cost offline.

III. Mean Field D2D Collaboration

In this section, we formulate a mean field model on graph to analyze D2D collaboration. Specifically, we will derive the state evolution by allowing the number of users N to approach infinity. Since users may move around, we consider the decentralized collaboration on both static and dynamic graphs to encompass the case of *time-varying D2D links*.

A. Collaboration on Static Graph

Basically, a static graph implies that D2D links are time-invariant, i.e., the graph $\mathcal{G}$ remains unchanged throughout the collaboration. To characterize state $\boldsymbol{s}$, we explore the transition of each $s_{k,i}$ from the perspective of a particular k-degree user u, as users are asymptotically independent when $N \to \infty$.

1) State evolution: Considering $s_{k,i}$ represents the workload distribution of k-degree users, then the transition events for the Markov chain include the following three instances:

- The number of tasks is $Q_u = i-1$, and u generates a task which stays at u, so the state transits from $s_{k,i-1}$ to $s_{k,i}$.
- The number of tasks is $Q_u = i-1$, and u receives a task sent from a neighbor, then $s_{k,i-1}$ transits to $s_{k,i}$.
- The number of tasks is $Q_u = i$, and u processes a task locally. Hence, the state changes from $s_{k,i}$ to $s_{k,i-1}$.

Transition probability of each instance is now provided. For the first instance, the polled neighbor by u must have no fewer tasks, which occurs with probability $q_{k,i-1}\sum_{k'\in\mathcal{K}} p(k'|k)s_{k',i} + q_{k,i-1}\frac{1}{2}\sum_{k'\in\mathcal{K}} p(k'|k)q_{k',i-1}$. The first term $q_{k,i-1}\sum_{k'\in\mathcal{K}} p(k'|k)s_{k',i}$ means $Q_u = i-1$ and the polled neighbor has at least i tasks. The second term $q_{k,i-1}\frac{1}{2}\sum_{k'\in\mathcal{K}} p(k'|k)q_{k',i-1}$ denotes tie breaking, namely the polled user also holds $i-1$ tasks. As for the second instance that u receives a task from a neighbor, its probability is $kq_{k,i-1}\sum_{k'\in\mathcal{K}} p(k'|k)s_{k',i}\frac{1}{k'} + kq_{k,i-1}\frac{1}{2}\sum_{k'\in\mathcal{K}} p(k'|k)q_{k',i-1}\frac{1}{k'}$, where $kq_{k,i-1}$ is because u has k neighbors and $\frac{1}{k'}$ implies that a k'-degree neighbor randomly polls u in Po2 choices (D2D collaboration) with probability $\frac{1}{k'}$. Similarly, the two terms represent situations where u has fewer tasks and tie breaks, respectively. The probability of the last instance is simply $q_{k,i}$. Combining task generation rate λ, offloading probability x and service rate of a mobile device μ, the state evolution is specified:

$$\begin{aligned}\dot{s}_{k,i} &= -\mu q_{k,i} + x_c\lambda q_{k,i-1}\sum_{k'\in\mathcal{K}} p(k'|k)\Big(s_{k',i} + \frac{1}{2}q_{k',i-1}\Big)\\ &+ kx_c\lambda q_{k,i-1}\sum_{k'\in\mathcal{K}}\frac{1}{k'}p(k'|k)\Big(s_{k',i} + \frac{1}{2}q_{k',i-1}\Big),\end{aligned} \tag{9}$$

where $x_c = 1-x$ is the probability that a task is processed via D2D collaboration and x would be $x[n]$ if in time slot n.

2) ODE system: Remember that $q_{k,i} = s_{k,i} - s_{k,i+1}$ and $\mathcal{G}$ is an uncorrelated graph. Based on Eq. (1), we have $p(k'|k) = \frac{k'p(k')}{\bar{k}}$. Therefore, for $i > 0$, Eq. (9) is simplified to:

$$\begin{aligned}&\dot{s}_{k,i} = -\mu(s_{k,i} - s_{k,i+1})\\ &+ x_c\lambda(s_{k,i-1} - s_{k,i})\Big[\frac{1}{2}\sum_{k'\in\mathcal{K}}\frac{k'+k}{\bar{k}}p(k')(s_{k',i-1} + s_{k',i})\Big].\end{aligned} \tag{10}$$

Besides, $s_{k,0} = 1$ according to Eq. (3). Define the drift function $\boldsymbol{F}(\boldsymbol{s}) = \{F_{k,i}(\boldsymbol{s})\}$, where $F_{k,0}(\boldsymbol{s}) = 0$ and $F_{k,i}(\boldsymbol{s}) = \dot{s}_{k,i}, \forall i > 0$. We have the following form:

$$\dot{\boldsymbol{s}} = \boldsymbol{F}(\boldsymbol{s}). \tag{11}$$

The *deterministic ODE system* of Eq. (10) corresponds to the *mean field model for our characterized D2D collaboration.*

B. Collaboration on Dynamic Graph

As users may move around, their neighbors within D2D communication range also change accordingly, resulting in a time-varying graph structure. Specifically, we leverage the model in [15], [16] to capture this dynamic feature.

1) Dynamic graph model: In a dynamic graph, each user has an expected degree k which is fixed and drawn from a finite set $\mathcal{K} = \{k_{\min}, ..., k_{\max}\}$ with probability $p(k)$. Value of expected degree indicates the willingness of a user to participate in collaboration. Hence, a D2D link between two users is established based on their expected degrees and spatial distance. In this regard, the number of D2D neighbors, or realized degree, of a user follows certain distribution conditioned on its expected degree k, which is specified as a Poisson distribution with the mean value being k in [16]. Due to user mobility, graph $\mathcal{G}(t)$ is dynamic with a time-varying edge set $\mathcal{E}(t)$, or changing realized degrees. Similar to the static case, $\mathcal{G}(t)$ formed by realized degrees is considered uncorrelated.

2) State evolution: Define $q_{k,i}, s_{k,i}$ as Eqs. (2)-(3), whereas k now denotes the *expected degree*. Similar to the static graph, transitions of state $s_{k,i}$ on dynamic graph also entail three instances. The main difference lies in the probability $p(k'|k)$, a user has expected degree k and its neighbor has expected degree k', is no longer the expression in Eq. (1). Instead, we use the conditional distribution of the realized degree to help compute this $p(k'|k)$. Due to space limit, we elucidate the details of deriving the state evolution in Appendix A in [29], and only present the final result here. We find out that the evolution of $s_{k,i}$ is exactly Eq. (10), i.e., *the mean field models on static and dynamic graphs are unified by the same ODE system.* Hereinafter, we will rely on Eq. (10) to analyze D2D collaboration in both static and dynamic scenarios as a whole.

C. Mean Field Model on Finite-degree Graphs

There have been many efforts devoted to mean field model on graphs, while existing works mostly focus on complete and infinite-degree graphs [13], [18], or apply the mean field analysis without theoretical guarantees [14], [19]. Therefore, our work has two novel contributions in the mean field aspect. First, we extend current mean field model on graphs to both *static and dynamic graphs with finite yet heterogeneous degrees*. Second, we also provide *rigorous proofs* (in the next section), along with extensive evaluations to demonstrate the effectiveness of our mean field model.

Now we discuss a special case where users have a homogeneous degree, that is the degree set $\mathcal{K} = \{k\}$ and $p(k) = 1$. Consider that the graph $\mathcal{G}$ is connected and uncorrelated. The mean field model becomes (irrelevant to degree k indeed):

$$\dot{s}_{k,i} = x_c\lambda(s_{k,i-1}^2 - s_{k,i}^2) - \mu(s_{k,i} - s_{k,i+1}). \tag{12}$$

This in fact degenerates to the classical Po2 result [13].

Our main objective is to obtain the stationary point $\boldsymbol{s}^*$ for state evolution, such that $\boldsymbol{F}(\boldsymbol{s}^*) = \boldsymbol{0}$. To this end, we will derive the existence and uniqueness of the stationary point, as well as demonstrating that the system state from *any initial point* will eventually converge to this stationary point.

IV. Stationary Point for State Evolution

In this section, we specify the stationary point for the mean field model to obtain the steady state of D2D collaboration.

A. Stationary Point

Since drift at the stationary point is 0, the MEC system is statistically stable, and hence we are able to achieve a tractable collaboration performance. For this reason, the existence and uniqueness issues of stationary point need to be explored.

1) Existence of stationary point: We first demonstrate that there exists a stationary point $\boldsymbol{s}^*$ for our mean field model. Considering $\boldsymbol{F}(\boldsymbol{s}^*) = \boldsymbol{0}$, we attain the existence by showing that the ODE system of Eq. (10) has a fixed point.

Theorem 1. *There exists a stationary point $\boldsymbol{s}^*$ for the mean field model.*

See Appendix B in [29] for the detailed proof. With the existence of stationary point $\boldsymbol{s}^*$, we still can not use $\boldsymbol{s}^*$ to directly represent the steady system state, as there may be multiple stationary points or the state $\boldsymbol{s}(t)$ may not converge to $\boldsymbol{s}^*$. This requires us to further address the problems of unique stationary point and convergence of state $\boldsymbol{s}(t)$.

2) Uniqueness and convergence: Due to the graph structure, mean field model also depends on the degree distribution, which makes it difficult to characterize the unique stationary point and state convergence. To circumvent this problem, we will show the uniqueness and convergence issues alternatively.

Coordinate-wise dominance. The state evolution of $s_{k,i}(t)$ is identified by the ODE of Eq. (10). To obtain the state convergence, we have to figure out how the initial values $\boldsymbol{s}(0)$ would influence the state $\boldsymbol{s}(t)$ at later time t. Define coordinate-wise dominance $\boldsymbol{s} \succeq \hat{\boldsymbol{s}}$ if $s_{k,i} \geq \hat{s}_{k,i}, \forall k \in \mathcal{K}, i \geq 0$. The lemma below states that the dominance at any time t is consistent with that of the initial values.

Lemma 1. *Let $\boldsymbol{s}(t)$ and $\hat{\boldsymbol{s}}(t)$ be the solutions to the ODE system of Eq. (10) at time t with the initial values being $\boldsymbol{s}(0)$ and $\hat{\boldsymbol{s}}(0)$, respectively. If $\boldsymbol{s}(0) \succeq \hat{\boldsymbol{s}}(0)$, then $\boldsymbol{s}(t) \succeq \hat{\boldsymbol{s}}(t)$.*

See Appendix C-A in [29] for the proof. With the dominance consistency, we demonstrate that every trajectory of the state converges to the stationary point in an appropriate metric.

Exponential convergence rate. To show convergence, we need to find a Lyapunov function $\phi(\mathbf{s})$ which satisfies: 1) $\phi(\mathbf{s})$ relates to the distance between $\boldsymbol{s}$ and $\boldsymbol{s}^*$; 2) $\phi(\mathbf{s})$ is strictly decreasing, except at $\boldsymbol{s}^*$. Here, $\phi(\mathbf{s})$ is constructed as:

$$\phi(\mathbf{s}) = \min_{\boldsymbol{s}^* \in \mathcal{S}^*} \sum_{i \geq 0} \frac{|\sum_{k \in \mathcal{K}} p(k)(s_{k,i} - s^*_{k,i})|}{2^i}, \tag{13}$$

where $\mathcal{S}^*$ is the stationary point set. To simplify $\phi(\mathbf{s})$, we denote $s_i = \sum_{k \in \mathcal{K}} p(k) s_{k,i}$ and $s_{(k),i} = \sum_{k \in \mathcal{K}} p(k) k s_{k,i}$. In line with Eq. (10), the evolution of s_i is specified as:

$$\dot{s}_i = \frac{x_c \lambda}{\overline{k}} \left(s_{i-1} s_{(k),i-1} - s_i s_{(k),i} \right) - \mu (s_i - s_{i+1}). \tag{14}$$

Using s_i, we have $\phi(\mathbf{s}) = \min_{\boldsymbol{s}^* \in \mathcal{S}^*} \sum_{i \geq 0} \frac{|s_i - s^*_i|}{2^i}$. The convergence of state $\boldsymbol{s}$ is derived by showing $\phi(\mathbf{s}) \to 0$.

Lemma 2. *If initial points $\boldsymbol{s}(0) \succeq \boldsymbol{s}^*, \forall \boldsymbol{s}^* \in \mathcal{S}^*$ or $\boldsymbol{s}^* \succeq \boldsymbol{s}(0), \forall \boldsymbol{s}^* \in \mathcal{S}^*$, $\boldsymbol{s}(t)$ converges to $\mathcal{S}^*$ with exponential rate.*

Please refer to Appendix C-B in [29] for the detailed proof. The exponential convergence rate in Lemma 2 is derived under certain initial conditions. However, if there is only one stationary point, global convergence from any initial point will be naturally obtained. This is because when there is a unique stationary point $\boldsymbol{s}^*$, $\boldsymbol{s}(t)$ always converges to $\boldsymbol{s}^*$ if $\boldsymbol{s}(0) \succeq \boldsymbol{s}^*$ or $\boldsymbol{s}^* \succeq \boldsymbol{s}(0)$. The dominance consistency in Lemma 1 then ensures the convergence of $\boldsymbol{s}(t)$ to $\boldsymbol{s}^*$ from other initial values.

Unique stationary point. Now we show that our mean field model has a unique stationary point, which will be proved by combining the dominance consistency property and the exponential convergence result.

Theorem 2. *There is a unique stationary point $\boldsymbol{s}^*$ for the mean field model.*

See Appendix C-C in [29] for the proof. Lemma 2 and Theorem 2 imply the result as stated in the following theorem.

Theorem 3. *The state evolution of D2D collaboration globally converges to a unique stationary point with exponential rate.*

B. Influence of Heterogeneous Degrees

In minimizing the system cost, one constraint is the collaboration fairness in Eq. (7b) pertaining to heterogeneous degrees. Concretely, a user with larger degree receives more tasks in D2D collaboration, meanwhile having a higher probability to forward tasks to its neighbors. This effect is presented below.

Theorem 4. *At the stationary point $\boldsymbol{s}^*$, larger-degree users tend to have heavier workloads. In other words, $s^*_{k,i} \geq s^*_{k',i}, \forall i \geq 0$ if $k > k'$.*

See Appendix D in [29] for the proof. This theorem reveals that users with heterogeneous degrees have *uneven workloads* because larger-degree users generally have higher contributions in D2D collaboration. Note that $s^*_{k,1}$ indicates the busy probability of a k-degree user with processing tasks. To avoid small-degree users free-ride large-degree neighbors, we bound the gap between $\max\{s^*_{k,1}\}$ and $\min\{s^*_{k,1}\}$ in the system cost minimization so as to ensure the collaboration fairness, where $\max\{s^*_{k,1}\}$ and $\min\{s^*_{k,1}\}$ now become $s^*_{k_{\max},i}$ and $s^*_{k_{\min},i}$, respectively, from Theorem 4. Fig. 2 compares typical values of $s^*_{k_{\max},i}$ and $s^*_{k_{\min},i}$ for some basic understanding.

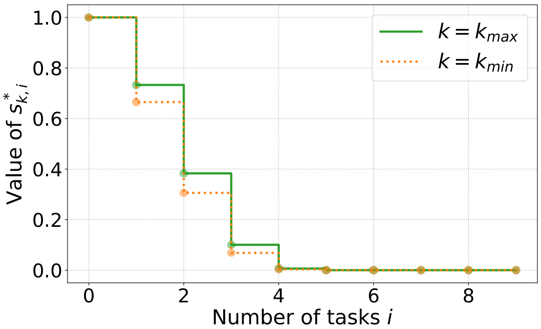

Fig. 2: Illustration of workload distribution.

C. Convergence to Mean Field Model

The mean field model makes use of a deterministic ODE system to analyze D2D collaboration in a stochastic MEC system. In the following, we show that the original stochastic N-user system will converge to the deterministic mean field model when N is large, namely the ODE system of Eq. (10) is accurate in describing the state evolution.

Consistent with Eq. (3), we denote $\boldsymbol{s}^{(N)}(t)$ as the state of N-user system. Our goal is to demonstrate that $\boldsymbol{s}^{(N)}(t) \to \boldsymbol{s}(t)$ when $N \to \infty$. To arrive at this conclusion, we first prove the drift function $\boldsymbol{F}(\boldsymbol{s})$ in Eq. (11) is Lipschitz continuous.

Lemma 3. *The drift function $\boldsymbol{F}(\boldsymbol{s})$ is $||\cdot||_\infty$-Lipschitz as there is a constant $C = 3x_c\lambda\big(1+\frac{k_{\max}}{\bar{k}}\big)+2\mu$ such that for any $\boldsymbol{s}, \hat{\boldsymbol{s}}$:*

$$||\boldsymbol{F}(\boldsymbol{s}) - \boldsymbol{F}(\hat{\boldsymbol{s}})||_\infty \leq C||\boldsymbol{s} - \hat{\boldsymbol{s}}||_\infty. \tag{15}$$

See Appendix E-A in [29] for the proof. With Lemma 3, we can claim the convergence of N-user system to the mean field model based on the Kurtz's theorem [20].

Theorem 5. *Fix a time t^*. When the number of users $N \to \infty$, the state $\boldsymbol{s}^{(N)}(t)$ converges to $\boldsymbol{s}(t)$ in the mean field model of Eq. (10) if they start from the same initial points.*

$$\lim_{N\to\infty} \sup_{t\in[0,t^*]} ||\boldsymbol{s}^{(N)}(t) - \boldsymbol{s}(t)||_\infty = 0, \text{a.s.} \tag{16}$$

Please refer to Appendix E-B in [29] for the proof. Theorem 5 ensures that the mean field model is effective for large population N. On this basis, one can obtain the existence and uniqueness of stationary point to characterize the steady state of MEC system and show the power of D2D collaboration.

D. Discussion of Stationary Point

1) Relation to classical Po2: The mean field models on static and dynamic graphs are unified by the ODE system of Eq. (10). When users have a homogeneous degree, Eq. (10) boils down to the classical Po2 of Eq. (12), which is independent of the degree and has a closed-form stationary point [13]:

$$\pi_i^* = \Big(\frac{x_c\lambda}{\mu}\Big)^{2^i-1}. \tag{17}$$

For a more general graph of heterogeneous degrees, an explicit expression for stationary point $\boldsymbol{s}^*$ is not available. Nevertheless, we can use the traditional Po2 as a bound for $\boldsymbol{s}^*$. Define two ratios:

$$\delta_1 = \frac{k_{\max}}{\bar{k}},\ \delta_2 = \frac{k_{\min}}{\bar{k}}. \tag{18}$$

Here, $\delta_1 > 1$ and $\delta_2 < 1$. Also, let $\frac{1+\delta_1}{2}\frac{x_c\lambda}{\mu} < 1$ to guarantee the system stability.

Corollary 1. *For the mean field model of Eq. (10), if $\frac{1+\delta_1}{2}\frac{x_c\lambda}{\mu} < 1$, then $s_{k,i}^*, \forall k, i$ has an upper bound:*

$$s_{k,i}^* \leq \Big(\frac{1+\delta_1}{2}\frac{x_c\lambda}{\mu}\Big)^{2^i-1}, \tag{19}$$

and a lower bound:

$$s_{k,i}^* \geq \Big(\frac{1+\delta_2}{2}\frac{x_c\lambda}{\mu}\Big)^{2^i-1}. \tag{20}$$

See Appendix F-A in [29] for the proof. According to this corollary, if the degree distribution is slightly heterogeneous, i.e., both the ratios δ_1 and δ_2 in Eq. (18) are close to 1, the gap between upper and lower bounds will be small, thereby leading to an accurate estimation of $s_{k,i}^*$ with closed-form expressions.

2) Busy probability: The probability that a user is busy with processing task, $s_1^* = \sum_{k\in\mathcal{K}} p(k)s_{k,1}^*$, is critical in computing the system cost in Eq. (5). The corollary below provides the value of s_1^* with the proof presented in Appendix F-B in [29].

Corollary 2. *The busy probability is $s_1^* = \frac{x_c\lambda}{\mu}$.*

3) Workload distribution relation: In the end of this section, we illustrate the relation between s_i^* and s_{i-1}^*. Based on Eq. (17), traditional Po2 satisfies $\pi_i^* = \frac{x_c\lambda}{\mu}\left(\pi_{i-1}^*\right)^2$. As for our mean field model on graph, similar conclusion is attained. Specifically, from Eq. (14), we know that the stationary point satisfies the following condition:

$$\frac{x_c\lambda}{\bar{k}}\left(s_{i-1}^* s_{(k),i-1}^* - s_i^* s_{(k),i}^*\right) - \mu(s_i^* - s_{i+1}^*) = 0. \tag{21}$$

Corollary 3. *For any $i \geq 1$, we have $s_i^* = \frac{x_c\lambda}{\bar{k}\mu}s_{i-1}^* s_{(k),i-1}^*$.*

See Appendix F-C in [29] for the proof. Compared to the classical Po2, the graph structure causes the difference between $\pi_i^* = \frac{x_c\lambda}{\mu}\pi_{i-1}^* \times \pi_{i-1}^*$ and $s_i^* = \frac{x_c\lambda}{\mu}s_{i-1}^* \times \frac{s_{(k),i-1}^*}{\bar{k}}$.

V. Online Offloading and Pricing Scheme

D2D collaboration can effectively reduce workloads of users. However, due to constrained resources of mobile devices, offloading a portion of tasks to a more powerful edge server is still essential to further mitigate task execution delay. Along with the offloading, there is a price charged by the edge server, so that users have to balance how many tasks should be offloaded and how many should be processed collaboratively. Task offloading between mobile users and edge server is modeled as a Stackelberg game, where the server is the leader in setting a service price which remains fixed for a long time, and users are followers in deciding the offloading probability.

A. Lyapunov Optimization

With *currently available information*, the server is interested in maximizing its long-term utility subject to the overloaded constraint by setting a proper price $p[n]$ in each time slot, as shown in Eq. (8). Meanwhile, mobile users aim to reduce their system cost while maintaining satisfactory task delay and collaboration fairness through determining the offloading probability $x[n]$, which is described in Eq. (7). To achieve these two goals, *Lyapunov optimization is leveraged to maximize the long-term utility online, with each time slot corresponding to a Stackelberg game to minimize the system cost offline.*

1) Optimal task offloading: At the beginning of time slot n, assume the edge server has declared a price $p[n]$. Users then collectively determine the offloading decision $x[n]$ to minimize their average system cost $c[n]$. Combining Eq. (5) and the busy probability in Corollary 2, we can rewrite $c[n]$ as:

$$c[n] = x[n]\lambda p[n] + (1-x[n])\lambda\frac{\rho_c^m}{\mu^2} + x[n]\lambda\rho_t^m\frac{B}{r}. \tag{22}$$

Critical points. Task delay $d(x[n])$ in Eq. (4) is composed of two parts. The first part is the transmission delay and completion time of task offloading, that is $d_o = x[n]\left(\frac{B}{r} + \frac{1}{\gamma}\right)$. The second part corresponds to the sojourn time of D2D collaboration $d_q = \frac{\sum_i \sum_k p(k) s^*_{k,i}}{\lambda}$. If $x[n]$ increases from 0 to 1, d_o will monotonically increase whereas d_q will monotonically decrease. Hence, there exist a lower bound x^*_l and an upper bound x^*_u such that when $x[n] \in [x^*_l, x^*_u]$ the delay constraint is fulfilled, where critical points x^*_l, x^*_u satisfy:

$$d\left(x[n] = x^*_l\right) = d\left(x[n] = x^*_u\right) = \overline{d}. \tag{23}$$

Note that the stationary point $\boldsymbol{s}^*$ is dependent on $x[n]$. We will compute each $s^*_{k,i}$ numerically given the value of $x[n]$, say $x[n] = x^*_l, x^*_u$, since there is no closed-form solution.

Let us discuss the collaboration fairness constraint, which is $s^*_{k_{\max},1} - s^*_{k_{\min},1} \le \overline{s}$ according to Theorem 4. The trend of the gap $s^*_{k_{\max},1} - s^*_{k_{\min},1}$ over $x[n]$ is not obvious. Nevertheless, if $x[n]$ approaches 1, i.e., users offload all tasks to the edge server, both $s^*_{k_{\max},1}$ and $s^*_{k_{\min},1}$ will be 0, and then their gap will be 0. When we push $x[n]$ approaching 0, that is users do not offload but only collaborate, all users will be heavily loaded, and hence $s^*_{k_{\max},1}$ and $s^*_{k_{\min},1}$ will be close to $\frac{\lambda}{\mu}$, with the gap being very small. As a result, we can characterize two feasible regions for $x[n]$: $[0, x'_l] \cup [x'_u, 1]$. If jointly considering the delay and fairness constraints, Fig. 3 provides a typical illustration of the feasible region for the offloading decision $x[n]$. In this paper, we assume that $[x^*_l, x^*_u] \cap ([0, x'_l] \cup [x'_u, 1]) \neq \emptyset$, namely $x[n]$ has a feasible solution. Additionally, for any feasible region, we denote x_l and x_u as its lower and upper boundary points, respectively, which hinge on the values of x^*_l, x^*_u, x'_l, x'_u, as also displayed in Fig. 3.

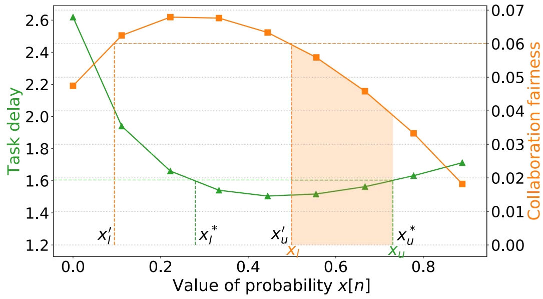

Fig. 3: Feasible region for offloading decision.

Threshold based offloading decision. To minimize the system cost in Eq. (22), provided the price $p[n]$, the optimal probability $x[n]$ for users is specified by a *threshold based decision*:

$$x[n] = \begin{cases} x_u & \text{if } \frac{\rho^m_c}{\mu^2} \ge p[n] + \rho^m_t \frac{B}{r}, \\ x_l & \text{otherwise.} \end{cases} \tag{24}$$

The intuition behind Eq. (24) is now explained. If the server sets an excessively high price such that the cost of offloading task is greater than the cost of processing task collaboratively, users will offload as few tasks as possible to reduce their system cost, and vice versa. With this reacted offloading decision $x[n]$, the edge server in turn determines the optimal price $p[n]$ in each time slot to maximize its service utility. Note that the critical points x_l, x_u are altered by $p[n]$ in that the price will affect the offloading decision. Recall from the overloaded constraint in Eq. (8a), it implies that $\overline{x} \in [x_l\lambda, x_u\lambda]$.

2) Dynamic service pricing: The server will judiciously choose the price over time to maximize its long-term utility, which follows an online Lyapunov optimization framework.

Drift-minus-utility. In view of the overloaded constraint in utility maximization, we define a virtual queue $X[n]$ for the edge server which buffers the virtual amounts of offloaded tasks. Here, we use the prefix "virtual" to denote that tasks are not actually offloaded from users, but rather, to reflect the requirement of the overloaded constraint. Consistent with this queue definition, tasks will enter into the queue with arrival rate $x[n]\lambda$ where $x[n]$ is users' offloading probability, and will leave the queue with departure rate $\overline{x}$. Therefore, we have the following dynamic equation for the virtual queue $X[n]$:

$$X[n+1] = \max(X[n] + x[n]\lambda - \overline{x}, 0). \tag{25}$$

Based on Eq. (25), we construct Lyapunov function as $\frac{1}{2}X^2[n]$, and compute Lyapunov drift which basically is the change of Lyapunov function from one time slot to the next:

$$\Delta(X[n]) = \mathbb{E}\Big[\frac{1}{2}X^2[n+1] - \frac{1}{2}X^2[n] \big| X[n]\Big]. \tag{26}$$

The expectation is taken over the randomness in task generation, offloading decision and pricing scheme. By minimizing Lyapunov drift $\Delta(X[n])$, one can drive the queue backlog to a small value so as to maintain $X[n]$ rate stable, $\lim_{n\to\infty} \frac{X[n]}{n} = 0$, with probability 1. From the queue stability theorem [21], a queue $X[n]$ is stable if and only if the arrival rate is no larger than the departure rate, i.e., $\lim_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \mathbb{E}[x[n]\lambda] \le \overline{x}$, and hence the overloaded constraint is satisfied. Furthermore, we define drift-minus-utility $\Delta(X[n]) - V\mathbb{E}[u[n]|X[n]]$, where V is the importance weight on the utility term. Minimizing drift-minus-utility will *simultaneously push the queue backlog to a small value and maximize the utility* as well [21].

Bound of drift-minus-utility. In time slot n, the queue $X[n]$ is known in advance. Besides, the offloading decision $x[n]$ is given in Eq. (24), which is also expressed as $x(p[n])$ to explicitly indicate its dependence on the price $p[n]$.

Lemma 4. *Drift-minus-utility $\Delta(X[n]) - V\mathbb{E}[u[n]|X[n]]$ satisfies the following condition:*

$$\Delta(X[n]) - V\mathbb{E}[u[n]|X[n]] \le \mathbb{E}\big[X[n](x(p[n])\lambda - \overline{x})|X[n]\big] \\ - \mathbb{E}\Big[\big(Vx(p[n])\lambda p[n] - Vx(p[n])\lambda\frac{\rho^s_c}{\gamma}\big)\big|X[n]\Big] + D, \tag{27}$$

where $D = \max(\frac{1}{2}\left(\lambda - \overline{x}\right)^2, \frac{1}{2}\overline{x}^2)$.

See Appendix G-A in [29] for the proof. Lemma 4 allows us to use a *simple-form bound* rather than the original complex drift-minus-utility in deriving the optimal price. As $X[n]$ is a priori knowledge in each time slot n and D is a constant, the remaining terms of Eq. (27) precisely correspond to the overloaded constraint and the service utility, respectively.

Optimal service price. In every time slot n, we minimize the bound of drift-minus-utility with $x(p[n])$ in Eq. (24):

$$\min\ X[n](x(p[n])\lambda - \overline{x}) - Vx(p[n])\lambda p[n] + Vx(p[n])\lambda\tfrac{\rho_c^s}{\gamma} \quad (28)$$

$$\text{s.t.}\ \ p[n] \in (0, p_u]. \quad (28a)$$

First, suppose that $p[n] \leq \frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}$, then $x(p[n]) = x_u$, thus the objective of Eq. (28) is $u_1(p[n]) = X[n]\,(x_u\lambda - \overline{x}) - Vx_u\lambda p[n] + Vx_u\lambda\frac{\rho_c^s}{\gamma}$. The price $p[n]$ should be $\frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}$ in order to minimize $u_1(p[n])$. Following this line, assume that $p[n] > \frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}$, and the objective becomes $u_2(p[n]) = X[n](x_l\lambda - \overline{x}) - Vx_l\lambda p[n] + Vx_l\lambda\frac{\rho_c^s}{\gamma}$, thereby $p[n]$ ought to be p_u. In general, we compare the two values $u_1(p[n] = \frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}) = X[n](x_u\lambda - \overline{x}) - Vx_u\lambda(\frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}) + Vx_u\lambda\frac{\rho_c^s}{\gamma}$ and $u_2(p[n] = p_u) = X[n](x_l\lambda - \overline{x}) - Vx_l\lambda p_u + Vx_l\lambda\frac{\rho_c^s}{\gamma}$ to determine $p[n]$. Let $X^* = \frac{Vx_u(\frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}) - Vx_l p_u}{x_u - x_l} - V\frac{\rho_c^s}{\gamma}$, then:

$$p[n] = \begin{cases} \dfrac{\rho_c^m}{\mu^2} - \rho_t^m \dfrac{B}{r} & \text{if } X[n] \leq X^*, \\ p_u & \text{otherwise.} \end{cases} \quad (29)$$

After setting price $p[n]$, offloading decision $x[n]$ is made based on Eq. (24), and queue $X[n]$ is updated following Eq. (25).

B. Performance of Offloading and Pricing Scheme

Now we provide the performance analysis of our task offloading and service pricing. In particular, we demonstrate that an asymptotically optimal utility is obtained and the queue backlog $X[n]$ has a constant upper bound.

Theorem 6. *Suppose the initial queue backlog $X[0] = 0$. For any importance weight $V > 0$, the proposed task offloading and service pricing satisfy the following properties.*

a) The queue backlog in any time slot n is bounded:

$$X[n] \leq \frac{Vx_u(\frac{\rho_c^m}{\mu^2} - \rho_t^m \frac{B}{r}) - Vx_l p_u}{x_u - x_l} - V\frac{\rho_c^s}{\gamma} + x_u\lambda - \overline{x}. \quad (30)$$

b) Denote u^* as the optimal time average utility for Eq. (8), then the achieved utility satisfies:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \mathbb{E}[u[n]] \geq u^* - \frac{D}{V}. \quad (31)$$

Please refer to Appendix G-B in [29] for the detailed proof. Theorem 6 unveils an $(O(V), O(1/V))$ tradeoff between the queue backlog and the service utility. Specifically, as the importance weight V increases, the queue backlog also increases as fast as the order of $O(V)$, while the time average utility approaches the theoretical optimum within an $O(1/V)$ gap. In addition, because $X[n]$ is upper bounded by a finite value given in Eq. (30), and then $X[n]$ is rate stable, that is the overloaded constraint will hold asymptotically.

VI. Performance Evaluation

In this section, we carry out evaluations to illustrate D2D collaboration among users and task offloading to edge server, especially to evaluate the mean field model on graph and Lyapunov optimization based offloading and pricing scheme.

A. Parameter Setting

In line with the real measurements [10], we set the average service time of a task to 1000 Megacycles which is normalized to 1 as stated in the system model, and the average data size B to 2000 KB. Service rates of a mobile device and the edge server are 1 GHz and 5 GHz, respectively, thus $\mu = \frac{10^9}{1000*10^6} = 1$ and $\gamma = \frac{5*10^9}{1000*10^6} = 5$ accordingly. Moreover, the typical real-world data rate of 4G cellular uplink is around $r = 10$ Mbps [22].

For D2D collaboration, task generation rate λ is set to 0.9 for modeling a heavy workload situation. User's (expected) degree in (dynamic) static graph is uniformly distributed in the set $\mathcal{K} = \{6, 7, 8, 9\}$, that is $p(k) = \frac{1}{4}, \forall k \in \mathcal{K}$ and $k_{\min} = 6, k_{\max} = 9$. Besides, user's realized degree given its expected degree obeys a Poisson distribution for dynamic graph [16], and the graph structure varies over time with rate 1.

Regarding task offloading, let $\overline{d} = 1.6$ for the delay constraint, and $\overline{s} = 0.06$ for the collaboration fairness constraint. The per energy costs are $(\rho_c^m, \rho_t^m, \rho_c^s) = (0.9, 0.3, 1)$, which correspond to 900 mW, 300 mW for processing and transmitting tasks in mobile devices, and 1000 mW for processing tasks in the edge server. The overloaded threshold $\overline{x}$ is set to 0.6, and the highest acceptable price p_u is assigned to 0.5.

B. Mean Field D2D Collaboration

Stationary point. We first demonstrate that the mean field model on graph is effective to characterize D2D collaboration by comparing the theoretical stationary point $\boldsymbol{s}^*$ obtained from the ODE system of Eq. (10) and that from simulating the Po2 choices. In particular, we compute the theoretical stationary point $\boldsymbol{s}^*$ using *scipy.integrate.odeint* in Python to solve the ODE system, since $\boldsymbol{s}(t)$ will converge to $\boldsymbol{s}^*$ when time t is large enough. On the other hand, the simulated MEC system consists of 800 users for static graph and 1000 users for dynamic graph. The static graph is generated by the configuration model [14] with both self-loops and multiple edges between two users being cut off to obtain an uncorrelated graph. Similarly, the configuration model is revoked when the graph structure changes to produce dynamic graph. Varying the value $x_c\lambda$, namely the proportion of tasks processed via D2D collaboration, from 0.1 to 0.9 with an increment of 0.1 each time, we run the simulated Po2 for eight times under each $x_c\lambda$. Figs. 4-5 exhibit the values of theoretical $s^*_{k_{\min},1}$ and averaged simulated $s^*_{k_{\min},1}$ on static and dynamic graphs, respectively, which tell that theoretical results perfectly match with simulated results. Therefore, the mean field model is effective in analyzing D2D collaboration.

Furthermore, we show each averaged simulated $s^*_{k,i}$ and theoretical $s^*_{k,i}$ when $x_c\lambda = 0.7$ in Table I. Results validate the accuracy of mean field model as the error between theoretical and simulated $s^*_{k,i}$ on static/dynamic graph is negligibly small.

User number influence. We continue discussing how the simulated Po2 behaves over time for different number of users N. In particular, we display the evolution of $s_{k_{\min},1}(t)$ when $N = 100, 300, 800$ for static graph and $N = 100, 300, 1000$ for dynamic graph in Figs. 6 and 7, respectively. We can see

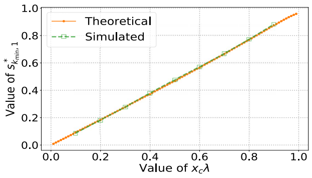

Fig. 4: Stationary point on static graph.

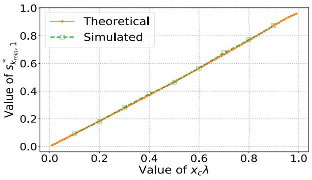

Fig. 5: Stationary point on dynamic graph.

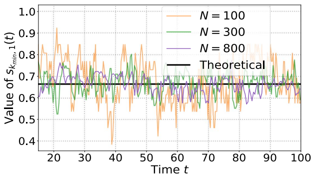

Fig. 6: Convergence to mean field model on static graph.

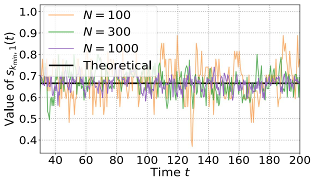

Fig. 7: Convergence to mean field model on dynamic graph.

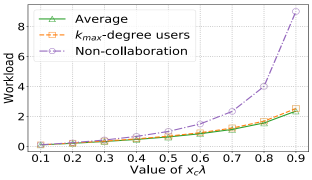

Fig. 8: Workload comparison.

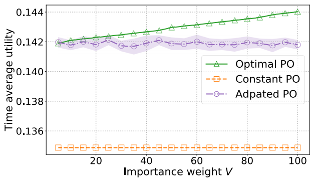

Fig. 9: Utility vs. V.

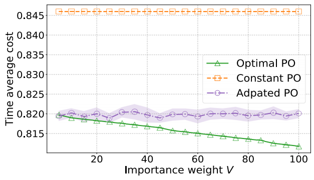

Fig. 10: Cost vs. V.

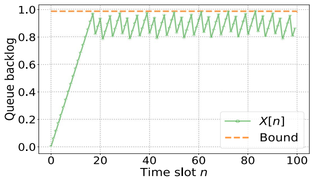

Fig. 11: Queue backlog $X[n]$.

TABLE I: Theoretical and simulated stationary points.

$s^*_{k,i}$	Theoretical	Static graph	Dynamic graph	Max error
$s^*_{6,1}$	0.66504	0.66706	0.67742	0.01238
$s^*_{7,1}$	0.68972	0.68924	0.696	0.00628
$s^*_{8,1}$	0.7123	0.71732	0.71786	0.00556
$s^*_{9,1}$	0.73295	0.73329	0.7325	0.00039
$s^*_{6,2}$	0.30585	0.31042	0.32056	0.01471
$s^*_{7,2}$	0.33239	0.34009	0.3492	0.01681
$s^*_{8,2}$	0.35814	0.35022	0.36905	0.00792
$s^*_{9,2}$	0.38302	0.36972	0.3852	0.0133

that as N increases largely, simulated results tend to approach the theoretical stationary point with impaired variances.

Power of collaboration. Theorem 4 reveals that largest-degree users have heaviest workloads. We now show that, even for those users, their workloads are still effectively mitigated in D2D collaboration, compared to the non-collaborative case which is a M/M/1 queue. Fig. 8 depicts the average workload and workloads for the largest-degree users as well as the non-collaborative case when value of $x_c\lambda$ varies. We can observe that the average and heaviest workloads are much smaller than the non-collaborative scenario, with the workload being mitigated by 73.8% when $x_c\lambda = 0.9$. Hence, the *task delay is also significantly reduced* as a result of the collaboration.

C. Lyapunov Optimization Based Offloading and Pricing

Critical points. We first derive the critical points for task offloading. As for the delay constraint, we implement golden section search to numerically compute critical points x^*_l, x^*_u by using Eq. (23), and hence we obtain $x^*_l = 0.26586, x^*_u = 0.72978$. Similarly, golden section search is leveraged to calculate critical points x'_l, x'_u for the fairness constraint, and their values are 0.08505 and 0.49953, respectively. Overall, the feasible region for the offloading decision is $[0.49953, 0.72978]$, namely $x_l = 0.49953, x_u = 0.72978$.

Utility and cost. Now we compare the performance of Lyapunov optimization (Optimal PO) with two baseline methods.

- Constant PO: edge server always chooses the Price p_u and users react with the Offloading probability x_l.
- Adapted PO: based on Eq. (29), edge server chooses the Price $\frac{\rho^m_c}{\mu^2} - \rho^m_t \frac{B}{r}$ or p_u with probability $\frac{\overline{x} - x_l\lambda}{x_u\lambda - x_l\lambda}$ or $\frac{x_u\lambda - \overline{x}}{x_u\lambda - x_l\lambda}$, and users react with the Offloading probability x_u or x_l.

It can be verified that the overloaded constraint also holds for these baseline methods. Let the total time slot $T = 100$, and we vary the importance weight V from 5 to 100 to obtain the corresponding service utility and system cost. Fig. 9 displays the time average utility, which shows that the utility will increase over V for Optimal PO as more emphasis is on the utility term. Besides, Optimal PO can achieve higher utility compared to Constant PO and Adapted PO. We then exhibit the time average system cost in Fig. 10, and we can see that Optimal PO leads to a lower cost than the baseline methods. With the increase of V, the edge server is more likely to set a lower price $\frac{\rho^m_c}{\mu^2} - \rho^m_t \frac{B}{r}$ from Eq. (29), that is why the system cost will decrease for Optimal PO. Therefore, Optimal PO can *achieve high energy efficiency* in optimizing utility and cost.

Queue backlog. Lastly, we depict the queue backlog $X[n]$ in Fig. 11 when the importance weight is $V = 20$. Results demonstrate that the queue backlog is finite under a constant bound derived in Theorem 6. This also implies the overloaded constraint in utility maximization is satisfied.

VII. Related Work

Collaborative MEC. Emerging MEC offers new possibility for intelligent mobile applications [1]. As single edge server has limited computing capacity, collaborative MEC is effective to accommodate more computation [23]. Li *et al.* propose an online learning aided collaborative offloading allowing edge servers to transmit tasks to each other based on a joint consideration of trust, delay and multi-hop transmission [24]. These works mainly focus on the collaboration among edge servers, instead of, among mobile users. To explore how users can help each other, Pu *et al.* study an incentive-aware task offloading among users via D2D links [5], but offloading to edge server is not considered. He *et al.* further incorporate D2D collaboration and task offloading to edge server for enhancing the computation capacity [8]. However, existing works rarely investigate D2D collaboration and task offloading as a whole. Besides, they mainly concentrate on the centralized collaboration for finite, often a handful of, users which makes them hard to be extended to large-population MEC systems.

Mean field model. Mean field model is used to characterize interactions among a large number of agents. Mitzenmacher uses the mean field model to analyze the power of d choices in randomized load balancing, where d queues are randomly sampled and a task will join the shortest queue [13]. Results show that even $d = 2$, the average sojourn time still decreases dramatically. Later on, Gast investigates the power of two choices on finite-degree graphs, but only simulation results are provided [19]. Budhiraja *et al.* show that the power of d choices on graphs can still be analyzed via mean field model, as long as each node has infinite degree [18]. Nevertheless, for D2D collaboration, the degree of a user (node) will not scale with the total number of users due to short-range D2D communication, so that users actually have finitely many neighbors. Another strand of researches using mean field model on graphs focus on epidemic processes in networks [14]. These works usually assume an uncorrelated graph, and use an ODE system to represent the state evolution, whereas the convergence to mean field model is often not proved [25]. Therefore, previous studies have not yet rigorously analyzed the mean field model on finite-degree or dynamic graphs.

Service pricing. Pricing scheme is important to a service provider when providing specific service for end users [26]. Regarding MEC, Zhao *et al.* propose a pricing scheme to charge mobile users when they offload computation via access points [27]. A Stackelberg game based heterogeneous pricing is designed to make decisions for edge servers (leaders) and mobile users (followers) in [28]. However, these works only consider finitely many users, and their pricing schemes are too complicated as uniform pricing is more easily implementable.

VIII. Conclusion

In this paper, we develop a joint D2D collaboration and task offloading for a large-population MEC system. Specifically, to characterize the state evolution of D2D collaboration, we propose a mean field model to analyze the stochastic MEC system by a deterministic ODE system. On this basis, we derive the existence and uniqueness of the stationary point, and further demonstrate the global convergence of state evolution to this unique stationary point. By incorporating D2D collaboration, we then design a pricing scheme for task offloading following a Lyapunov optimization framework. In particular, the offloading process is modeled as a Stackelberg game, where edge server is the leader to determine a reasonable price and users are followers to make the offloading decision. Extensive evaluations validate the effectiveness of our mean field model and the superiority of Lyapunov optimization.

Acknowledgments

This work is supported in part by the GRF 14201819 and CUHK:6905407.

References

[1] M. Satyanarayanan, "The Emergence of Edge Computing," *Computer*, vol. 50, no. 1, pp. 30-39, 2017.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE IoT*, vol. 3, no. 5, pp. 637-646, 2016.

[3] R. Kelly, "Internet of Things Data to Top 1.6 Zettabytes by 2020," Available: https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx

[4] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting Massive D2D Collaboration for Energy-Efficient Mobile Edge Computing," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 64-71, 2017.

[5] 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 JSAC*, vol. 34, no. 12, pp. 3887-3901, 2017.

[6] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint Task Assignment and Resource Allocation for D2D-Enabled Mobile-Edge Computing," *IEEE TCOM*, vol. 67, no. 6, pp. 4193-4207, 2019.

[7] X. Chen, Z. Zhou, W. Wu, D. Wu, and J. Zhang, "Socially-Motivated Cooperative Mobile Edge Computing," *IEEE Network*, vol. 32, no. 6, pp. 177-183, 2018.

[8] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D Communications Meet Mobile Edge Computing for Enhanced Computation Capacity in Cellular Networks," *IEEE TWC*, vol. 18, no. 3, pp. 1750-1763, 2019.

[9] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801-1819, 2014.

[10] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," *IEEE JSAC*, vol. 33, no. 12, pp. 2510-2523, 2015.

[11] L. Chen, S. Zhou, and J. Xu, "Computation Peer Offloading for Energy-Constrained Mobile Edge Computing in Small-Cell Networks," *IEEE/ACM TON*, vol. 26, no. 4, pp. 1619-1632, 2018.

[12] V. Cardellini, V. Personé, V. Valerio, F. Facchinei, V. Grassi, F. Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Mathematical Programming*, vol. 157, no. 2, pp. 421-449, 2016.

[13] M. Mitzenmacher, "The Power of Two Choices in Randomized Load Balancing," *IEEE TPDS*, vol. 12, no. 10, pp. 1094-1104, 2001.

[14] R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of modern physics*, vol. 87, no. 3, pp. 925-979, 2015.

[15] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-Varying Graphs and Dynamic Networks," *IJPEDS*, vol. 27, no. 5, pp. 387–408, 2012.

[16] J. C. Lang, H. D. Sterck, J. L Kaiser, and J. C. Miller, "Analytic models for SIR disease spread on random spatial networks," *Journal of Complex Networks*, vol. 6, no. 6, pp. 948-970, 2018.

[17] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," *IEEE/ACM TON*, vol. 24, no. 5, pp. 2795-2808, 2015.

[18] A. Budhiraja, D. Mukherjee, and R. Wu, "Supermarket Model on Graphs," [Online]. Available: https://arxiv.org/pdf/1712.07607.pdf

[19] N. Gast, "The Power of Two Choices on Graphs: the Pair-Approximation is Accurate," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 2, pp. 69-71, 2015.

[20] T. G. Kurtz, *Approximation of Population Processes*, Society for Industrial and Applied Mathematics, 1981.

[21] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool, 2010.

[22] 4G4U, "4G Speed Tests," Available: https://www.4g4u.org/4g-speed-tests/

[23] Z. Ning, P. Dong, X. Kong, and F. Xia, "A Cooperative Partial Computation Offloading Scheme for Mobile Edge Computing Enabled Internet of Things," *IEEE IoT*, vol. 6, no. 3, pp. 4804-4814, 2018.

[24] Y. Li, X. Wang, X. Gan, H. Jin, L. Fu, and X. Wang, "Learning-Aided Computation Offloading for Trusted Collaborative Mobile Edge Computing," *IEEE TMC*, vol. 19, no. 12, pp. 2833-2849, 2019.

[25] M. J. Farooq, and Q. Zhu, "Modeling, Analysis, and Mitigation of Dynamic Botnet Formation in Wireless IoT Networks," *IEEE TIFS*, vol. 14, no. 9, pp. 2412-2426, 2019.

[26] L. Zhang, W. Wu, and D. Wang, "Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes," in *Proc. IEEE INFOCOM*, 2014, pp. 700-708.

[27] Z. Zhao, W. Zhou, D. Deng, J. Xia, and L. Fan, "Intelligent Mobile Edge Computing With Pricing in Internet of Things," *IEEE Access*, vol. 8, pp. 37727-37735, 2020.

[28] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge," *Future Generation Computer Systems*, vol. 108, pp. 273-287, 2020.

[29] X. Wang, J. Ye, and John C.S. Lui, "Joint D2D Collaboration and Task Offloading for Edge Computing: A Mean Field Graph Approach," [Online]. Avaliable: https://arxiv.org/abs/2105.00766