Distributed Online Optimization of Fog Computing for Selfish Devices with Out-of-Date Information

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

Abstract—By performing fog computing, a device can offload delay-tolerant, computationally demanding tasks to its peers for processing, and the results can be returned and aggregated. In distributed wireless networks, the challenges of fog computing include lack of central coordination, selfish behaviors of devices, and multi-hop signaling delays which can result in outdated network knowledge and prevent effective cooperations beyond one hop. This paper presents a new approach to enabling cooperations of \( N \) selfish devices over multiple hops, where selfish behaviors are discouraged by a tit-for-tat mechanism. The tit-for-tat incentive of a device is designed to be the gap between the helps (in terms of energy) the device has received and offered; and indicates how much help the device can offer at the next time slot. The tit-for-tat incentives can be evaluated at every device by having all devices broadcast how much help they offered in the past time slot, and used by all devices to schedule task offloading and processing. The approach achieves asymptotic optimality in a fully distributed fashion with a time-complexity of less than \( O(N^2) \). The optimality loss resulting from multi-hop signaling delays and consequently outdated tit-for-tat incentives is proved to asymptotically diminish. Simulations show that our approach substantially reduces the time-average energy consumption of the state of the art by 50\% and accommodates more tasks, by engaging devices hops away under multi-hop delays.

Index Terms—Distributed fog computing, Selfish devices, Out-of-date information, Lyapunov optimization

I. INTRODUCTION

Computationally demanding mobile applications, such as face recognition, online gaming and eHealth, have been increasingly outgrowing the limited capability of individual mobile devices [1]. By performing fog computing, mobile devices can potentially help each other to tackle computationally demanding applications and tasks. A device can offload its tasks to other devices for processing, and the results can be returned and aggregated at the device [2]–[4].

It is non-trivial to optimally schedule fog computing in distributed wireless networks, due to the distributed nature and non-trivial topology of the networks, the randomness of mobile traffic and environments, and selfish nature of mobile devices [5]–[7]. All this can result in strong couplings of optimal computing and offloading schedules in time. Offloading tasks and results in non-trivial topologies can be NP-complete [8]. Tit-for-tat mechanisms can be designed to incentivize selfish devices to participate in fog computing [9]. However, the tit-for-tat incentive for one device to help another, generated by the latter, can be already out-of-date when reaching the former, as the result of multi-hop propagation delays. This would prevent effective cooperation of the devices.

To the best of our knowledge, these challenges have yet to be addressed in the literature. In a different yet relevant context of collaborative computing between data centers in a cloud, centralized off-line load balancing was studied to deal with the spatial diversities of workload arrivals [10], temperatures [11], and electricity prices [12]. In the context of mobile edge computing where devices are controlled by a single fog server, offloading decisions and resource allocations were typically optimized in a centralized manner [13]–[15]. For the sake of scalability, distributed scheduling of computation offloading was later proposed by applying game theory [16], [17], decomposition techniques [18], [19] and Lyapunov optimization [20].

A hierarchical fog architecture was considered in [21]–[25], where mobile devices can offload their tasks to either a fog server or a remote cloud. In [21], a computation offloading game was formulated to model the competition between devices for the limited computing resources. The game was proved to be a potential game, thereby confirming the existence of the Nash equilibrium. In [24], [25], different queuing models were applied to capture the delay performance of task processing at a mobile device, a fog node, and the cloud. In [24], a multi-objective optimization problem was formulated to obtain a Pareto-optimal solution among energy consumption, delay, and the payment to the fog service provider. In [25], each device aimed to minimize the execution cost of itself and its socially connected neighbors. In [22], [23], the synchronization of the cloud and edge devices was studied, and the end-to-end delays were analyzed with a focus on engineering practice.

In our previous work [26], the distributed online optimization of collaborative regions for fog computing among edge servers was developed without considering the selfish...
behaviors of mobile devices. In [27], [28], tit-for-tat incentives were used to discourage selfish behaviors of devices under the assumption that the incentives generated at individual devices can be instantly obtained at a central controller [27], [28]. The decisions of cooperations among devices were optimized given the up-to-date knowledge on the incentives by using Lyapunov optimization. Moreover, offloading was confined within a single hop with limited numbers of devices to cooperate [27], [28], hence preventing fog computing for large tasks.

This paper presents a new distributed online optimization of fog computing, where selfish devices are encouraged to cooperate over unrestricted numbers of hops to minimize the time-average energy consumption on delay-tolerant, computationally demanding tasks. By configuring $2N$ queues per device, the proposed approach enables all $N$ devices to collaborate with guaranteed return of results. Lyapunov optimization techniques are exploited to decouple the optimal decisions of devices on task processing and offloading, and result delivery. The contributions of the paper are beyond the direct application of Lyapunov optimization, and can be summarized as follows:

- The proposed approach optimizes cooperations of $N$ selfish devices over unrestricted numbers of hops. The approach is fully distributed, where every device optimizes its own schedule of task processing and offloading, and result delivery with a time complexity of less than $O(N^2)$. The schedules can collectively provide asymptotic optimality across the entire system.

- A new decentralized tit-for-tat mechanism is designed to incentivize selfish devices to collaborate. The tit-for-tat incentive of a device is the gap between the helps (measured by energy consumption) the device has received and offered; indicates how much help the device can offer at the next time slot; and can be evaluated at all devices based on simple signaling in which every device reports how much help it offered in the past time slot. With the tit-for-tat incentives of all other devices, each device can optimize its task processing and offloading based on the expected help from others.

- In the presence of multi-hop signaling delays, the aforementioned simple signaling can be delayed and the tit-for-tat incentives can be outdated. By taking the worst-case multi-hop signaling delays into account, we prove the optimality loss of the proposed approach resulting from outdated tit-for-tat incentives is upper bounded and can asymptotically diminish. The proposed approach preserves the asymptotic optimality under multi-hop signaling delays.

Corroborated by extensive simulations, the proposed approach is able to substantially reduce the time-average energy consumption of the state of the art by 50%, and accommodate significantly more tasks by engaging selfish devices hops away. In contrast, existing methods have been typically confined within a single hop under the assumption of up-to-date network knowledge [10]–[28]. Table I summarizes the comparison between the proposed approach and existing techniques.

The rest of this paper is organized as follows. In Section II, the system model is presented. In Section III, we propose the distributed online optimization of fog computing under negligible propagation delay. In Section IV, we extend the distributed online optimization under non-negligible propagation delay, where out-of-date incentives are interpreted as backlogs of virtual queues and the asymptotic optimality is proved. In Section V, the effectiveness of the proposed approach is demonstrated through extensive simulations, followed by conclusions in Section VI.

### II. System Model

Fig. 1 shows an illustrative example of distributed wireless networks supporting fog computing. The network under consideration consists of $N$ mobile devices. $\mathbb{N} = \{1, \cdots, N\}$ collects the indexes for the devices. The computational tasks...
generated by each of the mobile devices can be either locally executed, or offloaded to other devices via Device-to-Device (D2D) links for fog computing. Different from most existing assumptions, we allow a task to be offloaded multiple hops away. Once executed, the result of the task is returned to the device generating the task. The notations used in this paper are summarized in Table II.

The system run on a slotted basis with the slot length $T$. The devices can be synchronized by using existing time synchronization protocols, such as IEEE 1588 Precision Time Protocol (PTP) [29] and Timing-Sync Protocol for Sensor Networks (TPSN) [30]. Every pair of neighboring devices transmit to each other in sequel, exchange their timestamps of the transmissions and receptions, measure their round-trip delay, and calibrate their time offset. The devices can be synchronized with a typical offset of microseconds. By running TPSN, the time synchronization precision with errors of less than 2 $\mu$s can be stably achieved between two devices [31].

For a typical time slot of tens of milliseconds, such synchronization error of microseconds is negligible, and the time slots can be accurately configured among the devices. The schedule of time slots can be initialized by a nominated root device based on the clock time of the device. The devices within the one-hop range of the root device can synchronize with the root device by running TPSN. So on and so forth, all the devices can synchronize with the root device and configure the time slots accordingly.

We assume that the locations and wireless channels of the devices remain unchanged within a slot, and can change between slots. A stochastic graph $G(t) = \{N, E(t)\}$ is adopted to describe the topology of the network, where $N$ is represented by vertexes and $E(t) = \{(i, j)|e_{ij}(t) = 1, \forall i, j \in N\}$ collects edges $(i, j)$ between any pair of devices $i$ and $j$ with an active D2D link. $e_{ij}(t) = 1$ if devices $i$ and $j$ have a direct D2D link. Let $e_i(t)$ denote the energy consumption per bit for the transmission from device $i$ to $j$ at time slot $t$. It is upper bounded by $\epsilon_{ij}^{max}$, due to the finite transmit power of a device.

The capacity of the link during slot $t$, denoted by $C_{ij}(t)$, is assumed to be an independent and identically distributed (i.i.d.) stochastic process with an upper bound $C_{ij}^{max}$. $C_{ij}(t)$ can be observed by monitoring the transmit (Tx) buffer for link $(i, j)$ [32]. The re-transmission of data (i.e., tasks) is captured in $C_{ij}(t)$. This is because, by monitoring the Tx buffer for link $(i, j)$, $C_{ij}(t)$ can adapt to the time-varying wireless channel conditions and account for re-transmissions following failed transmissions. The D2D link can be asymmetric, i.e., $C_{ij}(t) \neq C_{ji}(t)$, due to a potential hidden node problem.

Let $F_i$ denote the computational capability of device $i$ (in CPU cycles per second), and $\delta_i(t)$ denote the percentage of background tasks at the device during slot $t$. Therefore, the available computational capacity of the device is $F_i(t) = 1 − \delta_i(t)F_iT$ for fog computing. Also let $e_i$ denote the energy consumption rate at device $i$ (in Joules per CPU cycle).

A computational task (such as face recognition, augmented reality, and e-health services), generated by device $i$, can be characterized by a triplet $(A_i(t), \rho_iA_i(t), \xi_iA_i(t))$ [27]. $A_i(t)$ is the size of the task in bits generated at slot $t$, it requires $\rho_iA_i(t)$ CPU cycles to process, and the result is $\xi_iA_i(t)$ bits. $\rho_i$ is the number of CPU cycles for processing a bit of the task, and $\rho_{min}$ and $\rho_{max}$ denote the minimum and maximum of the number, respectively. $\xi_i$ is the ratio of results to the corresponding unprocessed tasks in terms of size with the maximum $\xi_{max}$.

Note that $A_i(t)$ is an i.i.d. stochastic process with the maximum $A_{i}^{max}$. The assumption of i.i.d. task arrivals has been widely adopted in the literature [20], [24]–[28] and experimentally validated [33]. For example, the task arrivals at a mobile device, such as mobile applications and background services, are typically modeled as Poisson process [24], [25]. Within non-overlapping equal-length time windows (such as time slots of a slotted system), the numbers of occurrences of a Poisson process are known to be i.i.d. [34]. Hence, the number of Poisson task arrivals per time slot is i.i.d. [24], [25].

For illustration convenience, we consider divisible tasks, also known as the data-partition model [4, Section II-A-2], [15], [20], [24]–[26], such as gzip compression and feature extraction, for establishing the proposed approach and its asymptotic optimality. The tasks can be partitioned according to $C_{ij}(t)$ and $F_i(t)$ in (17) and (21). The proposed approach can also be applied to some statically partitioned tasks following the parallel dependency model [4, Fig. 4(b)], and preserve the asymptotic optimality, as will be discussed in Section III.

We propose that each device maintains $N$ queues for unprocessed tasks of all $N$ devices (including the device itself) and another $N$ queues for the corresponding results. The separate $2N$ queues enable all $N$ devices to collaborate and ensure the results to be returned to the device generating the task. Each of the queues is scheduled on a First-In-First-Out (FIFO) basis. As per slot $t$, $Q_i^{max}(t)$ denotes the backlog of the queue, in which device $i$ buffers unprocessed tasks originating from

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>The set of mobile devices</td>
</tr>
<tr>
<td>$T$</td>
<td>The slot length</td>
</tr>
<tr>
<td>$A_i(t)$</td>
<td>The size of tasks generated by device $i$ at slot $t$</td>
</tr>
<tr>
<td>$\rho_i$</td>
<td>The CPU cycles for processing a task bit of device $i$</td>
</tr>
<tr>
<td>$\xi_i$</td>
<td>The ratio of results to unprocessed tasks of device $i$ in size</td>
</tr>
<tr>
<td>$C_{ij}(t)$</td>
<td>The capacity of link $(i, j)$ during slot $t$</td>
</tr>
<tr>
<td>$e_{ij}(t)$</td>
<td>The energy consumption per bit of link $(i, j)$ during slot $t$</td>
</tr>
<tr>
<td>$F_i(t)$</td>
<td>The available computing resources of device $i$ at slot $t$</td>
</tr>
<tr>
<td>$\epsilon_i$</td>
<td>The energy consumption per CPU cycle of device $i$</td>
</tr>
<tr>
<td>$Q_{ij}(t)$</td>
<td>The task queue backlog at device $i$ for device $j$ at slot $t$</td>
</tr>
<tr>
<td>$D_{ij}(t)$</td>
<td>The result queue backlog at device $i$ for device $j$ at slot $t$</td>
</tr>
<tr>
<td>$E_i(t)$</td>
<td>The energy consumption on link $(i, j)$ during slot $t$</td>
</tr>
<tr>
<td>$E_i(t)$</td>
<td>The energy for processing tasks of device $i$ during slot $t$</td>
</tr>
<tr>
<td>$E(t)$</td>
<td>The overall energy consumption of all devices during slot $t$</td>
</tr>
<tr>
<td>$\theta_{ij}(t)$</td>
<td>The tasks of device $i$ offloaded from device $j$ at slot $t$</td>
</tr>
<tr>
<td>$d_{ij}^{max}(t)$</td>
<td>The results for device $i$ returned via device $j$ at slot $t$</td>
</tr>
<tr>
<td>$f_{ij}^{max}(t)$</td>
<td>The resources of device $i$ for device $j$'s tasks at slot $t$</td>
</tr>
<tr>
<td>$E_{ij}^{max}(t)$</td>
<td>The energy of device $i$ to help device $j$'s tasks at slot $t$</td>
</tr>
<tr>
<td>$C_i(t)$</td>
<td>The energy of device $i$ to help others at slot $t$</td>
</tr>
<tr>
<td>$S_i(t)$</td>
<td>The energy of other devices to help device $i$ at slot $t$</td>
</tr>
<tr>
<td>$Z_i(t)$</td>
<td>The hit-for-tat virtual backlogs of device $i$ at slot $t$</td>
</tr>
<tr>
<td>$Z_i^{max}(t)$</td>
<td>The out-of-date backlogs at device $i$ for device $j$ at slot $t$</td>
</tr>
<tr>
<td>$\epsilon_{ij}^{max}$</td>
<td>The diameter of stochastic connectivity graph over all slots</td>
</tr>
</tbody>
</table>
Device $s$ and $D_i^{(s)}(t)$ denotes the backlog of the queue, in which device $i$ buffers results destined for device $s$.

Fig. 2 illustrates the operations of an individual device in multi-hop fog computing, where there are two key decision gates. One of the decision gates is to decide the task to be executed based on the cost-effectiveness measure $\alpha_i^{(s)}(t)$, as to be given in (17). The other decision gate is to decide the transmission of unprocessed tasks or processed results over link $(i,j)$ based on the cost-effectiveness measures $\beta_{ij}^{(s)}(t)$ and $\gamma_{ij}^{(s)}(t)$, as to be given in (21) and (22), respectively.

We also design a tit-for-tat mechanism based on the time-average energy consumption (or in other words, the accumulative energy consumption). Let $E_i^{(s)}(t) = \epsilon_i f_i^{(s)}(t) + \sum_{r \not= i} c_{ir}(t) [b_{ir}^{(s)}(t) + d_{ir}^{(s)}(t)]$ be the energy that device $i$ consumes to help device $j$’s tasks, including those on task execution and transmission, at time slot $t$, $i \not= j$. Let $C_i(t)$ specify the energy consumption of device $i$ to help other devices at time slot $t$:

$$C_i(t) = \sum_{r \not= i} E_j^{(s)}(t).$$ (1)

Let $S_i(t)$ specify the energy consumption of other devices to help device $i$ at time slot $t:

$$S_i(t) = \sum_{r \not= i} E_j^{(s)}(t).$$ (2)

The tit-for-tat mechanism requires $S_i(t) = C_i(t)$, $\forall i \in N$, where $X(t) = \lim_{T \to \infty} \frac{1}{T} \sum_{\tau=0}^{T-1} E[X(\tau)]$ denotes the long-term time-average of $X(t)$. In this sense, how much help a device can get from others (measured by a unified metric of energy) depends on how much the device has offered to help others with their tasks.

Motivated by future helps from others when needed, a selfish device offers to help when it is lightly loaded and has spare resources. By this means, participations in fog computing are incentivized, and selfish behaviors are discouraged. From (1) and (2), device $i$ needs to have the knowledge on the energy consumption of other devices to update the tit-for-tat incentives of itself and the others for the decision gates of task processing, offloading, and result delivery in Fig. 2. The information needs to propagate hop by hop from the other devices until reaching device $i$. With the increase of network size, the information can be increasingly out-of-date due to multi-hop propagation delays, and so do the tit-for-tat incentives.

III. DISTRIBUTED ONLINE OPTIMIZATION OF FOG COMPUTING FOR SELFISH DEVICES

In this section, we present the new asymptotically optimal, distributed online optimization for the multi-hop offloading and processing of tasks with guaranteed result retrieval, provided the up-to-date information is always available at each device. As discussed in Section II, the tit-for-tat incentives can be out-of-date subject to multi-hop propagation delays. In Section IV, we will prove that the optimality loss due to the out-of-date tit-for-tat incentives is bounded, and can diminish to preserve the asymptotic optimality.

A. Problem Statement

In fog computing, energy consumption provides a unified measure for the cost of both computations and transmissions. Also, mobile devices are energy restrained. For these reasons, we aim to reduce the overall time-average energy consumption of fog computing. The overall energy consumption of all the devices at slot $t$, denoted by $E(t)$, can be written as

$$E(t) = \sum_{i\in N} \sum_{j\in N} E_{ij}(t) + \sum_{i\in N} E_i(t),$$ (3)

where $E_{ij}(t) = \epsilon_{ij} \sum_{s \in N} \{f_i^{(s)}(t) + d_{ij}^{(s)}(t)\}$ is the energy required to forward tasks and return results on link $(i,j)$ during slot $t$, and $E_i(t) = \epsilon_i \sum_{s \in N} f_i^{(s)}(t)$ is the energy required to execute tasks at device $i$ during the time slot.

Consider the stochastic nature of wireless channels, traffic arrivals and device mobility. We propose to minimize the overall time-average energy consumption of fog computing while stabilizing the queues of all the devices. The problem of interest can be formulated as

$$\mathbf{P}: \min_{B,B,B} \overline{E(t)}$$

s.t. $\mathbf{C1}: \sum_{s \in N} f_i^{(s)}(t) \leq F_i(t), \forall i \in N, t \in T$

$\mathbf{C2}: \sum_{s \in N} [b_{ij}^{(s)}(t) + d_{ij}^{(s)}(t)] \leq C_{ij}(t) \epsilon_{ij}(t) T,$

$\forall i,j \in N, t \in T$

$\mathbf{C3}: S_i(t) \leq C_i(t), \forall i \in N$

$\mathbf{C4}: f_i^{(s)}(t), b_{ij}^{(s)}(t), d_{ij}^{(s)}(t) \geq 0, \forall i,j,s \in N, t \in T$

$\mathbf{C5}: Q_i^{(s)}(t) < \infty, D_i^{(s)}(t) < \infty, \forall i,s \in N$ (4)

where $b_{ij}^{(s)}(t)$ specifies the size of unprocessed tasks originating from device $s$ and forwarded to device $j$ through device $i$ at slot $t$, $d_{ij}^{(s)}(t)$ specifies the size of results returned from device $i$ to $s$ through device $j$ at slot $t$, and $f_i^{(s)}(t)$ is the CPU cycles of device $i$ allocated to the tasks originating from device $s$ at slot $t$. Let $B = \{b_{ij}^{(s)}(t) | i,j,s \in N, t \in T\}$, $D = \{d_{ij}^{(s)}(t) | i,j,s \in N, t \in T\}$ and $F = \{f_i^{(s)}(t) | i,s \in N, t \in T\}$ collect these variables over all time slots $T = \{0,1,\cdots\}$.

Constraint $\mathbf{C1}$ ensures that the allocated CPU cycles do not exceed the available resources at any device and any time slot. $\mathbf{C2}$ restraints that the total transmit rate of both tasks and
results over a D2D link does not exceed the capacity of the link. C3 captures the tit-for-tat mechanism to motivate selfish devices to cooperate. C4 specifies that all the variables to be optimized are non-negative. C5 requires that all the queues at the devices are stable (i.e., the sizes of the queues must stay finite per slot).

Note that the optimal solution for $P$ would require the a-priori knowledge on task variations, channel fluctuations, mobility, and computing resources across all slots. On the other hand, myopic optimization of every slot would substantially increase the energy consumption since variables are coupled in time. In this paper, we decouple the variables between time slots, by converting $P$ to a Lyapunov optimization problem which can asymptotically minimize the time-average energy consumption while stabilizing the network (i.e., stabilizing the backlogs of the FIFO queues at all devices).

### B. Distributed Online Operations via Lyapunov Optimization

The Lyapunov optimization is a widely adopted stochastic optimization technique to decouple temporally and spatially coupled variables, derive asymptotically optimal solutions [35], and has been extensively used for smart grid [36], network function virtualization [37], as well as mobile edge computing [20]. The key idea of Lyapunov optimization is to minimize the upper bound of a drift-plus-penalty function per slot, and achieve an $[O(V);O(1/V)]$-tradeoff between the drift of Lyapunov function (i.e., the queue lengths) and the penalty (i.e., the time-average energy consumption to be minimized).

A quadratic Lyapunov function $L(t)$ can be defined to measure the stability of the queue, as given by [35]

$$
L(t) = \frac{1}{2} \sum_{i \in N} \left\{ Z_i(t)^2 + \sum_{s \in \mathcal{N}} \left[ Q_i^s(t)^2 + D_i^s(t)^2 \right] \right\}.  \tag{5}
$$

The Lyapunov function becomes large, as the queues move towards unstable states. The stability of the queues can be achieved by taking control actions that harness the Lyapunov function per slot [36].

In (5), we define a set of new virtual queues to capture the resource tit-for-tat constraint C3. At any device $i$ and time slot $t$, the backlog of the virtual queue, denoted as $Z_i(t)$, can be defined as [35]

$$
Z_i(t+1) = Z_i(t) + S_i(t) - C_i(t),  \tag{6}
$$

where $Z_i(0) = 0$, $\forall i \in N$. According to [35], the stability of $Z_i(t)$ is the sufficient condition for the satisfaction of C3. C3 can be relaxed by guaranteeing the stability of the virtual queues, i.e., $Z_i(t) < \infty$, $\forall i \in N$.

The backlog of unprocessed tasks $Q_i^{(s)}(t)$ is updated by

$$
Q_i^{(s)}(t+1) \leq \max\{Q_i^{(s)}(t) - f_i^{(s)}(t)/\rho_s - \sum_{j \in \mathcal{N}} b_{ij}^{(s)}(t), 0\} + \sum_{j \in \mathcal{N}} b_{ij}^{(s)}(t) + A_i^{(s)}(t),  \tag{7}
$$

where $f_i^{(s)}(t)/\rho_s$ is the number of bits that can be processed from the queue at time slot $t$, and $A_i^{(s)}(t)$ is the number of unprocessed tasks arriving at the queue at the time slot. $A_i^{(b)}(t) = A_i(t)$, and $A_i^{(s)}(t) = 0$, $\forall s \neq i$. The first term on the right-hand side (RHS) gives the remaining unprocessed tasks in the queue at the end of time slot $t$, after part of the tasks have been processed at device $i$ or offloaded to other devices. The other two terms give the new task arrivals offloaded from other devices or generated locally at device $i$.

The backlog of results $D_i^{(s)}(t)$ is updated by

$$
D_i^{(s)}(t+1) \leq \max\{D_i^{(s)}(t) - \sum_{j \in \mathcal{N}} d_{ij}^{(s)}(t), 0\} + \sum_{j \in \mathcal{N}} d_{ij}^{(s)}(t) + \xi_s f_i^{(s)}(t)/\rho_s, \forall s \neq i,  \tag{8}
$$

where $D_i^{(i)}(t) = 0$ since device $i$ is the sink for the results.

The inequalities in (7) and (8) are due to the fact that some devices may not have enough tasks or results to fulfill the rate that they accept. Nevertheless, the difference between the left-hand side (LHS) and the RHS of the inequalities marginalizes as tasks increase, and hence the queues saturate.

A drift-plus-penalty function can be defined to minimize the long-term energy consumption while stabilizing the system, as given by [35]

$$
\Delta_V(t) = \Delta(t) + V E[E(t)]  \tag{9}
$$

where $\Delta(t) \triangleq E[L(t+1) - L(t)]$ is the drift of the Lyapunov function, $E[E(t)]$ is the overall energy consumption at slot $t$, and $V$ is a predefined non-negative coefficient to tune the tradeoff between the queue lengths and energy consumption.

By minimizing the upper bound of $\Delta_V(t)$ per slot $t$, we can decouple the variables in time, stochastically minimize the time-average energy consumption, and stabilize the queues with asymptotic optimality. The upper bound of $\Delta_V(t)$ is established in the following Lemma.

**Lemma 1.** For any queue backlog and actions, $\Delta_V(t)$ is upper bounded by

$$
\Delta_V(t) \leq U + V E[E(t)] + \sum_{i \in \mathcal{N}} Z_i(t)E[S_i(t) - C_i(t)] + \sum_{i,j \in \mathcal{N}} Q_i^{(s)}(t)E[A_i^{(s)}(t) - f_i^{(s)}(t)/\rho_s + \sum_{j \in \mathcal{N}} (b_{ij}^{(s)}(t) - b_{ji}^{(s)}(t))] + \sum_{i,j \in \mathcal{N}} D_i^{(s)}(t)E[\xi_s f_i^{(s)}(t)/\rho_s + \sum_{j \in \mathcal{N}} (d_{ij}^{(s)}(t) - d_{ji}^{(s)}(t))],  \tag{10}
$$

where

$$
U = \frac{1}{2} \left\{ \sum_{i \in \mathcal{N}} (\xi_{s_{\max}} + 1)F_i T_{\rho_{\min}} + A_i^{\max} + 2 \sum_{j \in \mathcal{N}} C_{j_{\max}} T^2 \right\} + \sum_{i \in \mathcal{N}} \epsilon_i F_i T + \sum_{j \in \mathcal{N}} C_{j_{\max}} T^2.  \tag{11}
$$

**Proof.** Please see Appendix A.

Now, problem $P$ can be transformed to minimizing (10) at each time slot $t$, subject to the instantaneous constraints $C1$, $C2$ and $C4$. Let $b(t)$, $d(t)$ and $f(t)$ collect $b_{ij}^{(s)}(t)$, $d_{ij}^{(s)}(t)$ and $f_i^{(s)}(t)$, $\forall i, j \in \mathcal{N}$, respectively. By rearranging (10), $P$ can be rewritten as

$$
P1: \max_{b(t), d(t), f(t)} \quad g(f(t)) + \phi(b(t), d(t))
$$

s.t. $C1$, $C2$ and $C4$,

$$
g(f(t)) = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{N}} \{ f_i^{(s)}(t)Q_i^{(s)}(t)/\rho_s - D_i^{(s)}(t)\xi_s/\rho_s \}$

$$
- V\epsilon_i f_i^{(s)}(t) - Z_i(t)\xi_s f_i^{(s)}(t) - \epsilon_i f_i^{(s)}(t))\};  \tag{13}
$$

where
\[ \phi(b(t), d(t)) = \sum_{s, t \in \mathbb{N}} \left\{ -V \epsilon_{ij}(t) b_{ij}^{(s)}(t) + d_{ij}^{(s)}(t) \right\} \\
- Q_i^{(s)}(t)[b_{ji}(t) - b_{ij}^{(s)}(t)] - D_{ij}^{(s)}(t)[d_{ji}(t) - d_{ij}^{(s)}(t)] \\
- Z_i(t)[\epsilon_{si}(t)(b_{ji}^{(s)}(t) + d_{ij}^{(s)}(t)) - \epsilon_{ij}(t)(b_{ij}(t) + d_{ji}(t))]. \]

(14)

Both \( U \) and \( A^{(s)}(t) \) are independent of \( b(t) \), \( d(t) \) and \( f(t) \), and suppressed in \( P1 \).

Problem \( P1 \) can be efficiently solved by separately maximizing \( g(f(t)) \) subject to \( C1 \) and \( C4 \), as reformulated in \( P2 \), and \( \phi(b(t), d(t)) \) subject to \( C2 \) and \( C4 \), as reformulated in \( P3 \). This is because \( f(t) \) can be decoupled from \( b(t) \) and \( d(t) \) in both the objective and constraints of \( P1 \).

1) Computing Resource Allocation: From (13), the optimization of computing resources can be decoupled from each other. \( P2 \) is formulated to optimize \( f_i(t) = \{f_i^{(s)}(t), \forall s \in \mathbb{N}\} \) for each device \( i \), as given by

\[ \text{P2:} \max_{f_i(t)} \sum_{s \in \mathbb{N}} \alpha_i^{(s)}(t) f_i^{(s)}(t) \]

\[
s.t. \sum_{s \in \mathbb{N}} f_i^{(s)}(t) \leq F_i(t) \\
f_i^{(s)}(t) > 0, \forall s \in \mathbb{N} \]

where

\[ \alpha_i^{(s)}(t) = -V \epsilon_i + [Q_i^{(s)}(t) - \xi_s D_i^{(s)}(t)]/\rho_s + |Z_i(t) - Z_s(t)| \epsilon_i. \]

(16)

Apparently, \( P2 \) is linear programming in the form of weighted-sum maximization [38], where \( \alpha_i^{(s)}(t) \) serves as the weight for \( f_i^{(s)}(t) \). The optimal solution can be readily given by

\[ f_i^{(s)}(t) = \begin{cases} F_i(t), & \text{if } s = \arg \max_j \alpha_j^{(s)}(t) \text{ and } \alpha_i^{(s)}(t) > 0; \\ 0, & \text{otherwise.} \end{cases} \]

(17)

(16) and (17) indicate that all the available computing resources of device \( i \), \( F_i(t) \), are allocated to the queue with the most urgency or the most incentive to be processed.

2) Offloading and Routing Decisions: \( P3 \) is formulated to optimize the decisions of task offloading and result routing for each individual D2D link. This is because \( \phi(b(t), d(t)) \) can be decoupled between D2D links, and \( C2 \) specifies the capacity of each individual D2D link. Consider \( b_{ij}(t) = \{b_{ij}^{(s)}(t), \forall s \in \mathbb{N}\} \) and \( d_{ij}(t) = \{d_{ij}^{(s)}(t), \forall s \in \mathbb{N}\} \) for each D2D link \((i, j)\). \( P3 \) can be written as

\[ \text{P3:} \max_{b_{ij}(t), d_{ij}(t)} \sum_{s \in \mathbb{N}} \beta_{ij}^{(s)}(t)b_{ij}^{(s)}(t) + \gamma_{ij}^{(s)}(t)d_{ij}^{(s)}(t) \]

\[
s.t. \sum_{s \in \mathbb{N}} \left[ b_{ij}^{(s)}(t) + d_{ij}^{(s)}(t) \right] \leq C_{ij}(t) \epsilon_{ij}(t) T \\
b_{ij}^{(s)}(t), d_{ij}^{(s)}(t) \geq 0, \forall s \in \mathbb{N} \]

where

\[ \beta_{ij}^{(s)}(t) = -V \epsilon_{ij}(t) + [Q_i^{(s)}(t) - Q_j^{(s)}(t)] + |Z_i(t) - Z_s(t)| \epsilon_i(t); \]

(19)

\[ \gamma_{ij}^{(s)}(t) = -V \epsilon_{ij}(t) + [D_{ij}^{(s)}(t) - D_j^{(s)}(t)] + |Z_i(t) - Z_s(t)| \epsilon_i(t). \]

(20)

\( P3 \) is also a weighted-sum maximization like \( P2 \) [38], and its optimal solution can be obtained by evaluating the weights \( \beta_{ij}^{(s)}(t) \) and \( \gamma_{ij}^{(s)}(t) \). In specific, if \( \beta_{ij}^{(s)}(t) < 0 \) and \( \gamma_{ij}^{(s)}(t) < 0, \forall s \in \mathbb{N}, \) device \( i \) neither offloads unprocessed tasks originating from device \( s \), nor returns results destined for device \( s \), through device \( j \). If \( \max_{s \in \mathbb{N}} \{\beta_{ij}^{(s)}(t)\} > \max_{s \in \mathbb{N}} \{\gamma_{ij}^{(s)}(t)\} \), device \( i \) does not return results through device \( j \), i.e., \( d_{ij}^{(s)}(t) = 0, \forall s \in \mathbb{N} \), and the size of the unprocessed tasks that device \( i \) forwards to device \( j \) is given by

\[ b_{ij}^{(s)}(t) = \begin{cases} C_{ij}(t) \epsilon_{ij}(t) T, & \text{if } s = \arg \max_r \beta_{ij}^{(r)}(t); \\ 0, & \text{otherwise.} \end{cases} \]

(21)

Otherwise, if \( \max_{s \in \mathbb{N}} \{\beta_{ij}^{(s)}(t)\} \leq \max_{s \in \mathbb{N}} \{\gamma_{ij}^{(s)}(t)\} \), device \( i \) does not forward unprocessed tasks to device \( j \), i.e., \( b_{ij}^{(s)}(t) = 0, \forall s \in \mathbb{N} \), and the size of the results that device \( i \) returns through device \( j \) is given by

\[ d_{ij}^{(s)}(t) = \begin{cases} C_{ij}(t) \epsilon_{ij}(t) T, & \text{if } s = \arg \max_r \gamma_{ij}^{(r)}(t); \\ 0, & \text{otherwise.} \end{cases} \]

(22)

In this sense, given a link, the queue which can make the most use of the link at time slot \( t \) is activated to transmit unprocessed tasks or results.

3) Distributed Implementations: Both problems \( P2 \) and \( P3 \) are weighted-sum maximizations [38]. According to (17), (21) and (22), the queues with the maximum weights \( \alpha_i^{(s)}(t) \), \( \beta_{ij}^{(s)}(t) \) and \( \gamma_{ij}^{(s)}(t) \) are selected to be processed, or activated to occupy the link to transmit unprocessed tasks or results, respectively. In this sense, the weights are the cost-effectiveness measures to efficiently and asymptotically minimize the time-average energy consumption of fog computing. The selfish behaviors are discouraged by the cost-effectiveness measures, where a selfish device \( i \) would not receive help from another device \( j \), given \( \alpha_i^{(t)}(t) < 0, \beta_{ij}^{(t)}(t) < 0 \) and \( \gamma_{ij}^{(t)}(t) < 0 \) with the virtual queue \( Z_i(t) \gg 0 \).

From (17), (21) and (22), we can see that the proposed online optimization is distributed, where the optimal decisions can be made locally at every individual server based on the cost-effectiveness measures. Except the tit-for-tat incentives \( Z_i(t) \), the cost-effectiveness measures only require the local knowledge that a node can acquire from itself and its immediate neighboring nodes. Particularly, device \( i \) can acquire the queue backlogs of its immediate neighbor \( j \), i.e., \( Q_i^{(s)}(t) \) and \( D_{ij}^{(s)}(t) \), and the capacity \( C_{ij}(t) \) and cost \( \epsilon_{ij}(t) \) of the link between device \( i \) itself and device \( j \). This information can be instantly obtained as part of the process of the neighbor discovery protocol (NDP) [39].

Remark: The proposed optimization of fog computing has a sub-quadratic computational complexity of less than \( O(N^2) \).

Proof. At every time slot, each node \( i \) selects one queue out of its totally \( N \) task queues for local processing based on (17), with the computational complexity of \( O(N) \). The node also selects one of the \( 2N \) task and result queues for each of its \( K_i \) radio links for task offloading or result delivery based on (21) and (22), with the computational complexity of \( O(NK_i) \). Therefore, the overall computational complexity of the node is \( O(NK_i) \leq O(N^2) \) per time slot, since \( K_i \leq N \). This is much lower than the complexity of \( O(N^3) \) per time slot in the state of the art [27]. Moreover, given the \( K_i \) neighbors, the complexity of the proposed approach, \( O(NK_i) \), only grows linearly to the network scale \( N \) at any node \( i \). The scalability of the proposed algorithm is substantially improved, as compared to the state of the art [27].
4) Calculation of Tit-for-tat Incentives: Recall that $E_i^j(t)$ is the energy that device $i$ consumes to help device $j$’s tasks at time slot $t$, $i \neq j$, and $Z_s(t)$ is the tit-for-tat incentive of device $s$ and indicates how much energy the device can spend helping others at time slot $t$. $Z_s(t) = \sum_{\tau=0}^{t-1} \sum_{j \neq s} E_j^s(\tau) - \sum_{\tau=0}^{t-1} \sum_{j \neq s} E_j^s(\tau)$. $Z_s(t)$ is designed to be the difference between the total energy that the other devices have consumed to help device $s$ by time slot $(t-1)$, i.e., $\sum_{\tau=0}^{t-1} \sum_{j \neq s} E_j^s(\tau)$, and the energy that device $s$ has consumed to help others by time slot $(t-1)$, i.e., $\sum_{\tau=0}^{t-1} \sum_{j \neq s} E_j^s(\tau)$. Indicating how much help every device can offer at the upcoming time slot $t$, $Z_s(t)$, $\forall s$, is needed for every device $i$ to make optimal decisions on task processing and offloading, and result delivery, based on (16), (19) and (20).

To achieve this, every device is designed to broadcast a short message at every time slot, to report how much energy the device consumed to help another device at the last time slot. For example, device $s$ processes tasks for device $j$ ($i \neq s$) at the last time slot $(t-1)$, and reports the amount of energy it consumed on the tasks together with the identities of itself and device $j$. The message device $s$ sends is $\{E_j^s(1-t), s, j\}$. In the absence of multi-hop signaling delays, any device $i$ receives such message from every other device and updates its knowledge on the tit-for-tat incentives of all devices (including itself) for the upcoming time slot $t$, i.e., $Z_s(t) = Z_s(t-1) + \sum_{j \neq s} E_j^s(t-1) - \sum_{j \neq s} E_j^s(t-1), \forall s \in \mathbb{N}$.

In practice, the message from any device $s$ to device $i$ needs to be forwarded hop-by-hop, and can be significantly delayed. As a consequence, the tit-for-tat incentive $Z_s(t)$ can be outdated. In Section IV, we will prove that the optimality loss due to the outdated tit-for-tat incentives is upper bounded, and the asymptotic optimality of our approach can be preserved.

C. Performance Analysis

Let $E^*_s(t)$ denote the stochastically minimized time-average energy consumption of the entire system using distributed online optimization, and $E_{opt}$ be the energy consumption of P minimized offline (in a posteriori manner) by omitting causality. The gap between $E^*_s(t)$ and $E_{opt}$ can be shown to converge asymptotically with the growth of $V$, as revealed in the following theorem.

**Theorem 1.** The gap between $E^*_s(t)$ and $E_{opt}$ is within $U/V$, i.e., $E^*_s(t) \leq E_{opt} + U/V$. (23)

**Proof.** For brevity, the proof is suppressed here. Please refer to [35, Theorem 4.8] for the details. □

From Theorem 1, the proposed optimization of fog computing can asymptotically converge to the offline minimum that violates causality, as $V$ increases. In other words, the proposed approach is asymptotically optimal. We can further show the queue lengths are linear to $V$ in the following theorem.

**Theorem 2.** All queues of the proposed distributed fog computing are strongly stable and the time-average backlogs of the queues satisfy

$$\sum_{i, s \in \mathbb{N}} \overline{Q_i^s(t)} + \overline{D_i^s(t)} \leq U / \varepsilon + V (E^* - E_{opt}) / \varepsilon,$$  (24)

where $\varepsilon > 0$ and $E^* \geq E_{opt}$ satisfy the Slater condition [35].

**Proof.** Please see Appendix B. □

Theorems 1 and 2 reveal that there is an $[O(V), O(1/V)]$-tradeoff between the queue lengths and optimality gap. According to Little’s Theorem [34], the queuing delay is proportional to the total queue lengths in a queuing system. As a result, the tradeoff between the queue lengths and optimality loss can be translated to a tradeoff between the delay and energy consumption in this paper. The tradeoff can be adjusted by configuring the parameter $V$.

The proposed approach can also be used for some statically partitioned tasks following the parallel dependency model [4, Fig. 4(b)], which can be processed independently of each other. This model has also been widely adopted to study fog computing in the literature [13], [14], [16]–[19], [21], [27], [28]. The size of a statically partitioned task, denoted by $D_{task}$, does not have to be fixed, and the maximum size is denoted by $D_{max}$. The approach can be readily applied by rounding the allocated computing resources $F_i(t)$ and link schedules $C_{ij}(t)$ to the largest integer numbers of atomic tasks that can be supported, i.e., $C_i^{prop}(t) = D_{task}/\rho_k [F_i(t)]$ in (17) and $D_{ij}^{prop}(t) = D_{task} [C_{ij}(t)/D_{task}]$ in (21). The rounding does not violate the asymptotic optimality in Theorem 1. This is because the optimality loss due to the rounding are $\alpha_i^{(s)} \rho_k D_{max}$ and $\beta_i^{(s)} (t) D_{max}$, since the differences due to the rounding are at most $\rho_k D_{task} \leq \rho_k D_{max}$ CPU cycles and $D_{task} \leq D_{max}$ bits. The optimality loss is upper bounded by $O(1)$ by using (24), since $\alpha_i^{(s)}$ and $\beta_i^{(s)} (t)$ are linear to the product of $V$ and the queue backlogs, hence preserving the asymptotic optimality in Theorem 1. The time slot length $T$ can be adequately designed for efficient use of the links. For instance, the slot length can be set as $T = D_{max}/C_{min}$, so that at least one task can be transmitted per time slot. $C_{min}$ is the minimum capacity of an active link in the network.

IV. DISTRIBUTED OPTIMIZATION UNDER OUT-OF-DATE KNOWLEDGE

In this section, we consider the tit-for-tat incentives can be outdated due to the multi-hop signaling delays, since the message on how much energy a device consumes to help others at slot $t$ needs to be forwarded hop-by-hop, as dictated in Section III-B4. This is distinctively different from existing works under the assumption of instantaneous knowledge on the incentives at a central controller [27], [28].

By referring to [22], [23], the multi-hop signaling delay between devices $i$ and $j$, $\tau_{ij}$, can be given by

$$\tau_{ij} = \sum_{(i', j') \in P_{ij}} \tau_{i'j'}^{prop} + \tau_{i'j'}^{tx} + \tau_{i'j'}^{syn} + \tau_{i'j'}^{que},$$  (25)

where $\tau_{i'j'}^{prop}$, $\tau_{i'j'}^{tx}$, $\tau_{i'j'}^{syn}$ and $\tau_{i'j'}^{que}$ are the propagation delay, transmission delay, time synchronization error, and queuing delay over one-a-hop link $(i', j')$, respectively; and $P_{ij}$ is the route between devices $i$ and $j$.

In practice, priority is typically given to signaling to be placed head-of-line (HOL) for transmission. The signaling of interest is also small, only consisting of a few bytes indicating
Algorithm 1 The Proposed Distributed Online Optimization under Outdated Network Knowledge

For each device $i$ at the beginning of slot $t$:
1: Find neighboring devices and establish D2D pairs
2: Acquire $A_i(t)$, $F_i(t)$ and $\epsilon_{ij}(t)$, and queue backlogs, $Q_i^{(s)}(t)$ and $D_i^{(s)}(t)$
3: Exchange this information with the devices in D2D pairs
For each device $i$:
4: Evaluate the weights $\hat{\alpha}_{i}^{(s)}(t)$, $\hat{\beta}_{ij}^{(s)}(t)$ and $\hat{\gamma}_{ij}^{(s)}(t)$ by (16), (19) and (20), using the out-of-date backlogs $\hat{Z}_{s}^{(i)}(t)$
5: Allocate the computing resources according to (17)
6: Schedule the offloading and routing based on (21) and (22)
7: Virtual queue update at each device $i$
8: Update the out-of-date backlogs $\hat{Z}_{s}^{(i)}(t)$ by (28)

By comparing (28) to (6), we can see $\hat{Z}_{s}^{(i)}(t) = Z_s(t - \delta_{\text{max}})$. In the presence of multi-hop signaling delays, the asymptotic optimality of the proposed algorithm can be proved by evaluating the impact of the most severely outdated tit-for-tat incentives $Z_s(t - \delta_{\text{max}})$.

Using the out-of-date incentives $\hat{Z}_{s}^{(i)}(t)$, the out-of-date version of the cost-effectiveness measures, denoted by $\hat{\alpha}_{i}^{(s)}(t)$, $\hat{\beta}_{ij}^{(s)}(t)$ and $\hat{\gamma}_{ij}^{(s)}(t)$, can still be independently evaluated at each device $i$ using (16), (19) and (20). This does not violate the asymptotic optimality of the approach achieved under the assumption of up-to-date knowledge on incentives $Z_s(t)$ at every device, as will be shown in Theorem 3. The optimal decisions are based on the out-of-date cost-effectiveness measures. In particular, the computing resources are allocated to queue $Q_i^{(j)}$ with the maximum $\hat{\alpha}_{i}^{(j)}$ according to (17), and the queue with the maximum $\hat{\beta}_{ij}^{(s)}(t)$ and $\hat{\gamma}_{ij}^{(s)}(t)$ is activated to transmit unprocessed tasks and results over link $(i,j)$ based on (21) and (22). The proposed distributed optimization of fog computing under out-of-date knowledge is summarized in Algorithm 1.

We proceed to prove that the differences between the out-of-date and up-to-date tit-for-tat incentives are upper bounded at any time slot $t$, and so are the out-of-date and up-to-date cost-effectiveness measures. In turn, the optimality loss due to out-of-date knowledge is bounded at any time slot.

Lemma 2. At any slot $t$, the difference between $Z_s(t)$ and $\hat{Z}_{s}^{(i)}(t)$ is upper bounded by

$$|Z_s(t) - \hat{Z}_{s}^{(i)}(t)| \leq \delta_{\text{max}} \theta,$$

where $\theta = \sum_{i \in N}[\epsilon_i F_i + \sum_{i \in N} \epsilon_{ij} \epsilon_{ij} \epsilon_{ij} \epsilon_{ij} C_{ij}]$ is the maximum energy consumption of the network per slot, which depends on the size of the network $N$, and the upper bounds of the stochastic parameters, such as $F_i$, $\epsilon_{ij}$ and $C_{ij}$. ($F_i$, $\epsilon_{ij}$ and $C_{ij}$).

Proof. According to (6) and (28), the out-of-date backlog of a virtual queue at device $i$ satisfies

$$\hat{Z}_{s}^{(i)}(t) = Z_s(t - \delta_{\text{max}}).$$

The difference of $Z_s(t)$ between consecutive slots $t$ and $t + 1$, i.e., $|Z_s(t + 1) - Z_s(t)|$ accounts for the difference between the energy that device $s$ consumes to help others and the energy that other devices consume to help device $s$. The inequality $|Z_s(t + 1) - Z_s(t)| \leq \theta$ corresponds to an extreme case where all other devices consume all their energy to help device $s$ at slot $t$. Then, we can obtain $|Z_s(t) - \hat{Z}_{s}^{(i)}(t)| = |Z_s(t) - Z_s(t - \delta_{\text{max}})| = |\sum_{t+1}^{t+1} Z_s(t + 1) - Z_s(t)|$. Exploiting the inequality $|a + b| \leq |a| + |b|$, we can prove $|Z_s(t) - \hat{Z}_{s}^{(i)}(t)| = |\sum_{t+1}^{t+1} Z_s(t + 1) - Z_s(t)| \leq \sum_{t+1}^{t+1} |Z_s(t + 1) - Z_s(t)| \leq \delta_{\text{max}} \theta$. \hfill $\square$

Lemma 3. The differences between the instant weights and the weights based on the out-of-date information satisfy

$$|\hat{\alpha}_{i}^{(s)}(t) - \alpha_{i}^{(s)}(t)| \leq 2 \epsilon_i \delta_{\text{max}} \theta,$$

$$|\hat{\beta}_{ij}^{(s)}(t) - \beta_{ij}^{(s)}(t)| \leq 2 \epsilon_{ij} \delta_{\text{max}} \theta,$$

$$|\hat{\gamma}_{ij}^{(s)}(t) - \gamma_{ij}^{(s)}(t)| \leq 2 \epsilon_{ij} \delta_{\text{max}} \theta.$$

(31)
Proof. Based on (16), we have \(|\tilde{\alpha}_i^{(s)}(t) - \alpha_i^{(s)}(t)| \leq \\epsilon_i|\tilde{Z}_i^{(i)}(t) - Z_i^{(i)}(t)| - |Z_i(t) - Z_s(t)|\}. Exploiting the inequality \(|a - b| \leq |a - a| + |b - b|\), we have \(|\tilde{Z}_i^{(i)}(t) - Z_i^{(i)}(t)| - |Z_i(t) - Z_s(t)| \leq |\tilde{Z}_i^{(i)}(t) - Z_i(t)| + |Z_i(t) - Z_s(t)|\). Substituting (29) into the RHS of the last inequality, we can prove that \(|\tilde{\alpha}_i^{(s)}(t) - \alpha_i^{(s)}(t)| \leq 2\epsilon_i\delta max \theta\). Likewise, we can obtain \(|\tilde{\beta}_j^{(s)}(t) - \beta_j^{(s)}(t)| \leq 2\epsilon_j\delta max \theta\) and \(|\tilde{\gamma}_ij^{(s)}(t) - \gamma_j^{(s)}(t)| \leq 2\epsilon_j\delta max \theta\), based on (19) and (20), respectively.

We can show that the optimality loss of P1, stemming from the out-of-date knowledge, is bounded, as revealed in the following theorem.

**Theorem 3.** The optimality loss of the solution for P1, resulting from the out-of-date knowledge on Zs(t), is within a constant C, i.e.,
\[
g(f^*(t)) + \phi(b^*(t), d^*(t)) - g(f(t)) - \phi(b(t), d(t)) \leq C
\]
where \(\{f^*(t), b^*(t), d^*(t)\}\) is the solution to P1 in the presence of instantaneous knowledge, \(\{f(t), b(t), d(t)\}\) is the solution under out-of-date knowledge, and \(C = 4\delta max \theta |\sum_{i\in N} \epsilon_i F_i + \sum_{i,j\in N} \epsilon_{ij} C_{ij}^max| T\).

**Proof.** First we consider the optimality loss of P2, i.e.,
\[
g(f^*(t)) - g(f(t)).\]
From (13), we can obtain
\[
g(f(t)) = \sum_{i,s\in N} a_i^{(s)}(t) f_i^{(s)}(t) = \sum_{i,s\in N} (\alpha_i^{(s)}(t) f_i^{(s)}(t) + [\alpha_i^{(s)}(t) - \alpha_i^{(s)}(t)] f_i^{(s)}(t)).
\]
(32)

Since \(f(t)\) is the optimal solution under \(\tilde{\alpha}_i^{(s)}(t)\), and therefore, we can also obtain
\[
\sum_{i,s\in N} \alpha_i^{(s)}(t) f_i^{(s)}(t) = \sum_{i,s\in N} (\alpha_i^{(s)}(t) f_i^{(s)}(t) + [\alpha_i^{(s)}(t) - \alpha_i^{(s)}(t)] f_i^{(s)}(t))
\]
\[
= g(f^*(t)) + \sum_{i,s\in N} (\alpha_i^{(s)}(t) - \alpha_i^{(s)}(t)] f_i^{(s)}(t),
\]
(33)

where the second equality is due to the fact that \(g(f^*(t)) = \sum_{i,s\in N} \alpha_i^{(s)}(t) f_i^{(s)}(t)\).

Subtracting (33) from (32) and then substituting (31), we can obtain
\[
g(f^*(t)) - g(f(t)) \leq \sum_{i,s\in N} |\tilde{\alpha}_i^{(s)}(t) - \alpha_i^{(s)}(t)] f_i^{(s)}(t)) + f_i^{(s)}(t)\).
\]
(34)

Likewise, we can consider the optimality loss of P3 and prove
\[
\phi(b^*(t), d^*(t)) - \phi(b(t), d(t)) \leq 4\delta max \theta \sum_{i,j\in N} \epsilon_{ij} C_{ij}^max T.
\]
(35)

Adding up (34) and (35), we finally prove the theorem. □

From Theorems 1, 2, and 3, it can be readily established that the optimality loss, resulting from the out-of-date knowledge \(\tilde{Z}_s^{(i)}(t)\), is within \(\frac{U + C}{V}\), i.e.,
\[
E^{opt}(t) \leq E^{opt} + \frac{U + C}{V}.
\]
(36)
The time-average backlogs of the queues satisfy
\[
\sum_{i,s\in N} Q_i^{(s)}(t) + D_i^{(s)}(t) \leq \frac{U + C}{\epsilon} + \frac{V(E^* - E^{opt})}{\epsilon}.
\]
(37)

That is to say, the distributed optimization under out-of-date knowledge is still asymptotically optimal, with an \([O(V), O(1/V)]\)-tradeoff.

**V. SIMULATION AND NUMERICAL RESULT**

In this section, we evaluate the proposed distributed online optimization of fog computing with \(N = 20\) selfish devices in the presence of non-negligible multi-hop propagation delays in a custom-designed Matlab platform. Our Matlab simulator is focused on the allocation of physical resources for computing and transmission (in CPU cycles and bits), since any allocations of the so-called virtual resources would have to be translated to physical resource requirements and implemented physically [4]. The Matlab simulator allows for fair comparisons of the proposed approach to the state of the art [27] which was evaluated on a custom-designed platform. Particularly, we assume that all tasks are of the same type, and every device is equipped with a single processor, as assumed in [27]. The key step of the state of the art is [27, Eq. 18], which can be reliably solved by using the Matlab optimization toolbox.

We run 10000 slots simulations for each data point, with the slot duration \(T = 1\) sec. The CPU capacity of the mobile devices \(F_i\) is uniformly distributed from 1 to 4 GHz with \(\epsilon_i = 1/F_i\) Joule/cycle, the background task \(\delta_i(t)\) uniformly and randomly varies from 0 to 40%, and the D2D transmission power is 0.2 W for each link. These simulation parameters are drawn from the dataset [27] for fair comparisons with the state of the art. The propagation delay is linear to the transmission hops, i.e., 0.2 sec/hop, and the connectivity ratio among devices \(p_c\) is set to be 0.25. The tasks arrive at devices following a binomial distribution with the probability 0.2, where \(A_i(t)\) is uniformly distributed in [1, 8] Mbits, \(\rho_i \in [1000, 3000]\) cycles/bit, and \(\xi_i = 1\).

For comparison purpose, we also simulate two benchmark approaches: (a) local execution, where the devices buffer and execute all the tasks locally; and (b) D2D Fogging [27], where the devices can offload tasks to direct neighbors which immediately execute the tasks and then return the results all within a time slot.

Fig. 3 compares the stabilized system throughput and energy efficiency between the proposed approach and the existing benchmarks, where the task size \(A_i(t)\) increases from 0.5 to 4.5 Mbits and \(V = 70\). In Fig. 3(a), we can see that the proposed approach and local execution process all arrived tasks and therefore achieve the same throughput when the tasks are small, i.e., \(A_i(t) < 2.5\) Mbits. However, as the task size increases, the tasks cannot be all processed locally, and the growth of the system throughput slows down under local execution. In contrast, the proposed approach is still able to process all tasks through multi-hop offloading. In Fig. 3(b), we can see that the proposed approach provides higher energy efficiency (up to 50% energy saving) than local execution and D2D Fogging when \(A_i(t) = 0.5\) Mbits, since much more tasks can be accomplished through fog computing over multiple hops. However, the gap of the proposed approach decreases with the growth of the task sizes, due to the increasing
requirement of offloading tasks further away and subsequently the increasing demand on energy.

In Fig. 3, we can see that traffic reshaping is important for the schemes that need to complete the offloading and processing of tasks, and the return of results within a slot, for example, D2D Fogging [27]. Despite the fact that D2D Fogging is susceptible to the task size and its throughput declines with the growth of the task size, it can provide higher energy efficiency for small tasks than the proposed approach. This is because the increase of task size can increasingly result in long processing delays, hence violating the requirement of completing a task within a slot and leading to the decrease of throughput. In this case, the devices that experience good link conditions and abundant computing resources are likely to satisfy the requirement and therefore offload tasks through energy-efficient transmission, thereby leading to the increased energy efficiency of D2D Fogging in Fig. 3(b).

Fig. 4 evaluates the stabilized system throughput and energy efficiency of the proposed approach, as the connectivity ratio $p_c$ increases, where $V = 70$ and $A_i(t) = 4.5$ Mbits. In Fig. 4(a), we can see that the proposed approach can provide 2Mbps more throughput than the local execution, as the connectivity improves. Particularly, when $p_c \leq 0.1$, tasks are prone to be offloaded over multiple hops and processed there under the proposed approach, increasing the throughput and preventing unbounded growth of the queues, as shown in Fig. 4(a). In contrast, the throughput of the local execution is limited by the computing capability of individual devices, and the queues of some devices can grow unrestrainedly. The additional energy consumption of offloading in Fig. 4(b) contributes to the increase of throughput and the stability of the queues, as compared to the local execution.

Fig. 5 plots the stabilized system throughput and energy consumption of the proposed approach, as $V$ increases from 0 to 150. As shown in the figure, once stabilized, the system throughput is always equal to the task arrival of all the devices, i.e., $0.2 \times 4.5 \times 20 = 18$ Mbps. On the other hand, the stabilized energy consumption of the proposed distributed fog computing decreases with $V$, due to the increasing priority on the minimization of energy consumption in $\Delta_V(f)$. However, the decline of energy consumption slows down and the energy consumption becomes stable when $V > 50$. This is the energy required to stabilize of all queues in the network.

Fig. 6 shows the average backlogs of tasks and results, as $V$ increases from 0 to 150. The backlogs continue increasing with $V$ for both types of task queues. However, the average backlog of the result queues is much shorter than that of the task queues. Moreover, the increase of the average backlog of results is quite slow when $V \leq 30$. This is because the

---

**Fig. 3.** The comparing of the stabilized system throughput and energy efficiency between the proposed approach and the existing benchmarks, as $A_i(t)$ increases from 0.5 to 4.5 Mbits.

**Fig. 4.** The stabilized system throughput and energy efficiency of the proposed approach versus connectivity ratio, where $V = 70$ and $A_i(t) = 4.5$ Mbits.

**Fig. 5.** The stabilized system throughput and energy consumption versus $V$. 
The result queues

The result queues

12

14

Backlog Length (Mbit)

200

100

50

10

30

40

60

90

Average Task Backlog (Mbit)

30

Virtual Queue Backlog (Joule)

0

1

2

4

6

7

Average Result Backlog (Mbit)

Fig. 8. The change of the backlog of virtual queues over time.

Fig. 7. The change of the average backlog over time.

Fig. 6. The average backlog of the task and result queues versus $V$.

increase of the result queues relies on the processing of the task queues, as can be seen in (8). Figs. 5 and 6 validate the tradeoff between the optimality gap and queue lengths established in Theorems 1 and 2, and corroborate the bounded optimality loss in Theorem 3.

Fig. 7 shows the change of the average backlog of the queues in time, where $t$ increases from 0 to 10000. We can see that both the backlogs of the task and result queues stabilize, and the average backlogs (i.e., the delay of the queues to be stabilized) increase with $V$, which is consistent with Fig. 6. Particularly, in Fig. 7(b), the time delay before the backlogs of the result queues start increasing also increases with $V$. This is because the increasing emphasis on the minimization of energy consumption requires the difference between the task and result queues to be enlarged for processing.

Fig. 8 shows the change of the backlog of virtual queues of 4 devices, $Z_i(t), i = 1, \ldots, 4$, as $t$ elapses, where $V = 70$. The backlogs of the virtual queues fluctuate drastically when $t \leq 2000$, and become stabilized when $t > 2000$. Particularly, the backlogs stabilize to different values. This difference is due to the heterogeneity of the devices (here, $F_i$ and $\rho_i$ are randomly initialized), i.e., the devices with higher computing capacities are more likely to help execute other devices’ tasks, and therefore achieve lower backlogs of their virtual queues. The stabilized virtual queue backlogs at the devices validate the effectiveness of the proposed tit-for-tat incentives, i.e., a selfish device cannot request services from others without offering (otherwise, the virtual queue backlogs would keep increasing and cannot be stabilized).

VI. CONCLUSION

In this paper, the processing and offloading of tasks, and the return of results were jointly optimized for multi-hop wireless fog computing in the presence of out-of-date signaling resulting from multi-hop propagation. Lyapunov optimization techniques were exploited to asymptotically minimize the time-average energy consumption with an $[O(V), O(1/V)]$-tradeoff between the optimality gap and delay. The impact of out-of-date signaling on the asymptotic optimality was proved to be bounded, hence preserving the asymptotic optimality of the proposed approach. Simulations show that our approach is able to reduce the time-average energy consumption by 50% and accommodate larger tasks by engaging wireless devices hops away to collaborate.

The proposed approach is developed under the assumptions that all tasks are of the same type, and can either be divisible or statically partitioned under the parallel dependency model [4, Fig. 4(b)]. The assumptions hold in many application scenarios and have been extensively adopted in the literature [13], [14], [16]–[19], [21], [27], [28]. Yet, there are other scenarios where there are multiple different types of tasks, and the tasks can follow the general dependency model [4, Fig. 4(c)] and need to be processed in particular orders, such as those modeled by directed acyclic graphs (DAG) [40]. Non-trivial effort would be required to extend the proposed approach to those scenarios, such as automated installation of virtual instances (at different devices) for different task types and optimal offloading of dependent DAG tasks. These will be the focus of our future research, with preliminary results in [37] based on sequentially dependent tasks and unselfish devices.

APPENDIX A

Taking squares on both sides of (7) and (8), and then exploiting the identity inequality for $(\max[a - b, 0] + c)^2 \leq a^2 + b^2 + c^2 + 2a(c - b)$ for any $a, b, c \geq 0$, we can obtain

\begin{equation}
\begin{aligned}
&[Q_i(t + 1)]^2 - [Q_i(t)]^2 \\
&\leq [f_i(t)/\rho_i + \sum_{j \in N} b_{ij}(t)]^2 + [\sum_{j \in N} b_{ij}(t) + A_i(t)]^2 \\
&+ 2Q_i(t)\sum_{j \in N} (b_{ij}(t) - b_{ij}(t)) + A_i(t) - f_i(t)/\rho_i,
\end{aligned}
\end{equation}

\begin{equation}
\begin{aligned}
&[D_i(t + 1)]^2 - [D_i(t)]^2 \leq \sum_{j \in N} d_{ij}(t) + \xi_j f_i(t)/\rho_i \\
&+ [\sum_{j \in N} d_{ij}(t) + 2D_i(t)\xi_j f_i(t)/\rho_i + \sum_{j \in N} (d_{ij}(t) - d_{ij}(t))].
\end{aligned}
\end{equation}

From (6), we have

\begin{equation}
Z_i(t + 1)^2 - Z_i(t)^2 \leq C_i(t)^2 + S_i(t)^2 + 2Z_i(t)[S_i(t) - C_i(t)].
\end{equation}
Plugging (38)–(40) into (9), we can achieve (10) with $U$ satisfying

$$
U \geq \frac{1}{2} \sum_{i \in N} \{E[f_i(s)(t)/\rho_s + \sum_{j \in N} b_{ij}(s)(t)^2] + E[\sum_{j \in N} d_{ij}(s)(t)^2]
+ E[\sum_{j \in N} (b_{ij}(s)(t) + A_i(s)(t))^2] + E[\sum_{j \in N} (d_{ij}(s)(t) + \xi_s f_i(s)(t)/\rho_s)^2]\}
+ \frac{1}{2} \sum_{i \in N} E[C_i(t)^2] + E[S_i(t)^2].
$$

(41)

Exploiting the inequality $(\sum_i a_i)^2 \geq \sum_i a_i^2$ for all $a_i \geq 0$, (11) provides an upper bound for (41) by replacing the expectations with their maximums.

**APPENDIX B**

By referring to [35, Theorem 4.8], there exist $\varepsilon > 0$, $E^\varepsilon \geq E^{opt}$, and the corresponding control policy $\pi^\varepsilon$ in the feasible region of the problem of interest $P$, such that [35, Eqs. (4.61)-(4.64)]

$$
E[E(t)/\pi^\varepsilon] = E^\varepsilon,
$$

$$
E[S(t) - C_i(t)/\pi^\varepsilon] \leq 0,
$$

$$
E[A_i(s)(t) - f_i(s)(t)/\rho_s + \sum_{j \in N} b_{ij}(s)(t) - b_{ij}(t)/\pi^\varepsilon] \leq -\varepsilon,
$$

$$
E[\sum_{j \in N} (d_{ij}(s)(t) - d_{ij}(t)/\pi^\varepsilon)] \leq \varepsilon.
$$

(42)

By plugging (42) into (10), we have

$$
\Delta_V(t) \leq U + V E^\varepsilon - \varepsilon \sum_{i,s \in N}[Q_i(s)(t) + D_i(s)(t)],
$$

(43)

where the RHS is a constant term $(U + V E^\varepsilon)$ minus a positive term $\varepsilon \sum_{i,s \in N}[Q_i(s)(t) + D_i(s)(t)]$, $U$ is obtained in Lemma 1. As compared to [35, Eq. (3.20)], $V E^\varepsilon$ is an additional pre-configurable constant, accounts for the penalty in the drift-plus-penalty (9), and can asymptotically diminish as $V$ decreases. As a result, the queue backlogs $Q_i(s)(t)$ and $D_i(s)(t)$ increase, $\varepsilon \sum_{i,s \in N}[Q_i(s)(t) + D_i(s)(t)]$ becomes increasingly large. The RHS of (43) decreases and becomes negative, and so does the Lyapunov drift-plus-penalty $\Delta_V(t)$. This feature results in the Lyapunov stability and the boundlessness of the optimal solution.

By reorganizing, (43) can be rewritten as

$$
\varepsilon \sum_{i,s \in N}[Q_i(s)(t) + D_i(s)(t)] \leq U + V E^\varepsilon - V E[E(t)]
- E[L(t + 1) - L(t)].
$$

(44)

Taking expectations on both sides of (44) and summing up the telescoping series over $\{0, 1, \cdots, T - 1\}$, we have

$$
\frac{1}{T} \lim_{T \to \infty} \sum_{t=0}^{T-1} \sum_{i,s \in N} \left[E[Q_i(s)(t)] + E[D_i(s)(t)]\right]
\leq \frac{U}{\varepsilon} + \frac{V(E^\varepsilon - E^{opt})}{\varepsilon} + \frac{E[L(0)] - E[L(T)]}{\varepsilon T}
\leq \frac{U}{\varepsilon} + \frac{V(E^\varepsilon - E^{opt})}{\varepsilon},
$$

where $E^{opt} \leq E[E(t)]$ is the offline optimum, and the second inequality is due to the fact that $E[L(T)] \geq 0$ and $L(0) = 0$. This concludes the proof of Theorem 2.

**REFERENCES**


Xinchen Lyu received the B.E. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently pursuing Ph.D. degrees at BUPT and University of Technology Sydney. His research interests include stochastic network optimization with applications to fog computing, software-defined networking, edge caching and radio resource management.

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

Dr Ni serves as Editor for Hindawi Journal of Engineering since 2012, secretary of IEEE NSW VTS Chapter since 2015, Track Chair for VTC-Spring 2016 and 2017, and Publication Chair for BodyNet 2015. He also served as Student Travel Grant Chair for WPMC 2014, Program Committee Member of CHINACOM 2014, TPC member of IEEE ICC’14, ICC’15, EIICE’14, and WCNC’10.

Hui Tian received her M.S. in Micro-electronics and Ph. D degree in circuits and systems both from Beijing University of Posts and Telecommunications, China, in 1992 and 2003, respectively. Currently, she is a professor at BUPT, the Network Information Processing Research Center director of State Key Laboratory of Networking and Switching Technology and the MAT director of Wireless Technology Innovation Institute (WTI). Her current research interests mainly include radio resource management, cross layer optimization, M2M, cooperative communication, mobile social network, and mobile edge computing.

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

Professor Liu is the founding chair of IEEE NSW VTS Chapter and a Senior Member of IEEE. He served as Technical Program Committee chair and Organising Committee chair in a number of IEEE Conferences. Ren Ping Liu received his B.E.(Hon) and M.E. degrees from Beijing University of Posts and Telecommunications, China, and the Ph.D. degree from the University of Newcastle, Australia.
Xin Wang received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree from Auburn University, Auburn, AL, USA, in 2004, all in electrical engineering. He was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, from 2004 to 2006. In 2006, he joined the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA, where he is an Associate Professor. He is currently a Distinguished Professor with the Department of Communication Science and Engineering, Fudan University. His research interests include stochastic network optimization, energy-efficient communications, cross-layer design, and signal processing for communications.

Georgios B. Giannakis received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1981, and the M.Sc. degrees in electrical engineering and mathematics and Ph.D. degree in electrical engineering from the University of Southern California, in 1983 and 1986, respectively. He was with the University of Virginia from 1987 to 1998, and since 1999 he has been a Professor with the University of Minnesota, where he was an Endowed Chair in Wireless Telecommunications, a University of Minnesota McKnight Presidential Chair in ECE, and serves as the Director of the Digital Technology Center. His general interests span the areas of communications, networking and statistical signal processing subjects on which he has published more than 400 journal papers, 680 conference papers, 25 book chapters, two edited books and two research monographs (h-index 119). Current research focuses on learning from big data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-)inventor of 28 patents issued, and the (co-)recipient of eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a fellow of the EURASIP and has served the IEEE in a number of posts, including that of a distinguished lecturer for the IEEE-SP Society. He is a fellow of the IEEE.

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