DOTS: Delay-Optimal Task Scheduling among Voluntary Nodes in Fog Networks

Guowei Zhang, Fei Shen, Nanxi Chen, Pengcheng Zhu, Xuewu Dai, and Yang Yang, *Fellow, IEEE*

***Abstract*—Through offloading the computing tasks of the task nodes (TNs) to the fog nodes (FNs) located at the network edge, the fog network is expected to address the unacceptable processing delay and heavy link burden existed in current cloud-based networks. Unlike most existing researches based on the command-mode offloading and full capability report, this paper develops a general analytical model of the task scheduling among voluntary nodes (VNs) in fog networks, wherein the VNs voluntarily contribute their capabilities for serving their neighboring TNs. A novel Delay-Optimal Task Scheduling (DOTS) algorithm is proposed to obtain the delay-optimal offloading solution according to the reported capabilities of the VNs. Extensive simulations are carried out in a fog network, and the numerical results indicate that the proposed DOTS algorithm can effectively provide the optimal helper nodes (HNs) set, subtask sizes, and TN transmission power to minimize the overall task processing delay. Moreover, compared with the command-mode offloading, the voluntary-mode achieves more balanced offloading and a higher fairness level among the FNs.**

***Index Terms*—Fog network, voluntary capability report, delay minimization, fairness.**

I. Introduction

WITH THE development of Internet of Things (IoT) technology, billions of resource-limited devices are expected in future networks [2], [3], and most of them are connected to the Internet [4], [5]. Due to the limited processing capability at each single node, tasks of most terminal devices need to be offloaded to achieve satisfactory delay performance for emerging applications, such as autonomous vehicles, smart home devices, e-healthcare, intelligent manufacturing, etc. In traditional networks, those tasks are usually offloaded to the cloud server, thus avoiding long processing delay and quick energy consumption at local terminal devices with limited capabilities [6]. Nevertheless, the explosive growth of 5G / IoT applications and the corresponding mobile data will generate heavy burden to the cloud server and all wireless links, thus the overall system performance will suffer dramatic degradation [7], [8]. Even worse, the long distances between the cloud server and different terminal devices cannot effectively support the delay-sensitive tasks [9], [10]. Therefore, more flexible and efficient task scheduling architectures and schemes are required to optimize the end-to-end delay performance by using fog-enabled task scheduling technology.

In fog-enabled systems, massive fog nodes (FNs) are distributed across the whole network [11], [12]. Aided by the advanced technologies including software defined network (SDN) and network function virtualization (NFV) [13], communication, computing, relaying, caching, and control services can be flexibly deployed on these ubiquitous FNs. The network resources are extended from the central cloud to the FNs, and this novel network structure provides a rich collection of ubiquitous computing, communication, and storage resources across the network [14]. The FNs holding varying capabilities can be jointly scheduled to achieve better system performances in delay, energy consumption, etc. Benefiting from the huge amount and the flexible deployment of the FNs in a fog network, the computing tasks generated at the task nodes (TNs) can be divided into multiple subtasks and offloaded to several nearby FNs rather than the remote cloud server. Therefore, task scheduling services with better quality of service (QoS) than traditional cloud computing can be provided by the effective utilization of available capabilities and resources in the neighborhood [11], [12], [14]. These heterogeneous FNs are either specifically deployed by the network operators or by autonomous clients with network-connected computing devices, which could voluntarily contribute their available resources for serving their neighboring TNs. However, the following three key questions need to be addressed.

- Under what conditions will an FN contribute its available capability and serve as a voluntary node (VN)?
- How does a VN determine how much capability it contributes for the task scheduling service?
- How to determine the delay-optimal task scheduling solution based on the capabilities contributed collectively from the nearby VNs?

In order to provide better support for delay-sensitive services in fog networks, extensive researches has been focused on the delay performance under different fog computing frameworks [15]–[20]. In [16], the delay-minimization problem is formulated as an integer optimization problem based on a hierarchical architecture named Combined Fog-Cloud (CFC). Fog computing is introduced as a complement to cloud

This research is partially supported by the National Science and Technology Major Project under grant 2017ZX03001015, the Natural Science Foundation of Shanghai, China under grant 18ZR1437500, and the Hundred-Talent Program of Chinese Academy of Sciences under grant Y86BRA1001. Part of this work has been presented at IEEE WCSP, Hangzhou, China, 2018 [1].

Guowei Zhang, Fei Shen, and Nanxi Chen are with Shanghai Institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences (CAS), Shanghai 200050. Guowei Zhang is also with University of Chinese Academy of Sciences, Beijing 100049, China. Pengcheng Zhu is with Southeast University, Nanjing 211189, China. Xuewu Dai is with Northumbria University, Newcastle upon Tyne, UK. Yang Yang is with ShanghaiTech University, Shanghai 201210, China, and also with Shanghai Institute of Fog Computing Technology (SHIFT), Shanghai, China.

Corresponding author: Fei Shen (fei.shen@mail.sim.ac.cn)

computing and an essential ingredient of the IoT in [17]. Analytical model is introduced for service delay based on queueing theory, and the delay-minimized policy is provided. In [18], the optimization problem is decomposed into two operations at the control tier and the access tier to respectively determine the node assignments and bandwidth allocation, and tradeoff is achieved between average network throughput and service delay. The authors of [19] provided a dual focus on the improvements of both the processing delay and the transmission delay, which are addressed respectively by virtual machine migration and transmission power control. A Fog-Radio Access Network (F-RAN) architecture is introduced in [20] by Shih *et al.* to guarantee the requirements of those ultra low-latency applications, wherein the mobile augmented reality (AR) service is taken as an example to illustrate the designed framework. Taking both the delay performance and the energy consumptions into consideration, many researchers focus on the problem of delay-energy tradeoff in fog-enabled systems [21]–[27]. The energy consumptions in the offloading processes are minimized in single-user [21] and multiple-user [22] scenarios respectively. Yang *et al.* [23] introduced a control parameter to characterize the delay-energy tradeoff during dynamic task scheduling processes in fog networks, and they also provided an effective algorithm to minimize the overall energy consumption while reducing average service delay and delay jitter. The performance indexes including task delay and energy consumption are abstracted to revenue and cost in the operation process of the fog-enabled computing network [24], [25], and game theories are adopted to achieve the balance of payments. The energy efficiency of task scheduling is maximized in [26], and the task is processed within a fixed period. In [27], three queueing models are adopted respectively for mobile terminal devices, the fog, and the cloud centers. A multi-objective optimization problem is formulated to minimize the energy consumption, execution delay, and payment cost by finding the optimal offloading probability and transmission power for each terminal device.

The existing task scheduling policies are almost all in the command-mode. In other words, all the FNs and their idle capabilities can be selflessly contributed to the TN to fulfill the task scheduling service. However, quite a portion of the FNs in real fog networks want to autonomously determine how much capability they contribute to nearby TNs according to their own current states. We call this more realistic scenario as the voluntary-mode. Furthermore, the FNs may take part in the offloading service in a totally economy-mode and contribute their capabilities according to the price that the TN offers. The voluntary-mode or economy-mode are to be the normalcy of the future fog networks. On the one hand, these modes can incent more network nodes to contribute more idle capabilities, thus more flexible and high-quality services can be provided. On the other hand, in a fog network under command-mode, most of the tasks of the TN are offloaded to the FNs which are close to the TN and possess high computing capabilities. This will lead to the halts of some important FNs due to the excessive consumption of energy, and thus some serious system-level problems, which could be avoided by the self-protection of the FNs in the voluntary-mode fog networks. To this end, we keep a watchful eye on the scheduling of the delay-sensitive tasks among the VNs in fog networks, and the three key questions raised above are going to be investigated.

In particular, the main contributions of this paper are summarized as follows.

- A general analytical model in a fog network under the voluntary-mode is constructed. An FN that voluntarily contributes its capability serves as a VN and otherwise a non-voluntary node (N-VN), and the delay-sensitive TN tasks can be offloaded to multiple selected VNs, named helper nodes (HNs). A report ratio is introduced and modeled for each FN to autonomously determine whether it serves as a VN, and the computing capability that the VN reports to the TN. The FNs with high historical energy consumptions will decrease their report ratios and thus report capabilities. Thus, the first two key questions raised above are addressed.
- The minimization problem for overall task processing delay of the task scheduling service among the VNs is formulated, and a Delay-Optimal Task Scheduling (DOTS) algorithm is proposed. The delay-optimal offloading solution is achieved by transforming the delay minimization problem to the equivalent rate maximization problem, and the optimal solution includes the set of the HNs, subtask sizes, and the TN transmission power to each HN. Thus, the third key question raised above is addressed.
- The performance of the proposed algorithm is evaluated through extensive simulations in a fog network. Numerical results demonstrate that the DOTS algorithm can effectively determine the optimal task scheduling solution that minimizes the overall task processing delay. Moreover, supported by the voluntary capability report, balanced offloading and fair energy consumptions among FNs are achieved by the proposed DOTS algorithm.

The rest of this paper is organized as follows: The task scheduling model in a voluntary-mode fog network and the corresponding problem formulation for the minimization of task delay are provided in section II. In Section III, the capability report ratio is introduced and modeled. Section IV provides the DOTS algorithm. In Section V, we conduct the simulations and performance analyses. Section VI concludes this paper.

II. System Model and Problem Formulation

In this section, we introduce the model of the task scheduling service among VNs in a fog network, and the task delay is formulated as the function of the offloading solution. A fog network as shown in Fig. 1 is considered, which consists of N FNs. Denote the FN set by $\mathcal{F}$. The N heterogeneous FNs have varying inherent characteristic parameters including central processing unit (CPU) cycles for processing 1 bit data denoted by η_i, the CPU frequency denoted by f_i, the energy consumption per CPU cycle denoted by θ_i, and etc. Each FN uses a bandwidth W for the data transmission with other network nodes. The key notations in this paper are provided in Table. I.

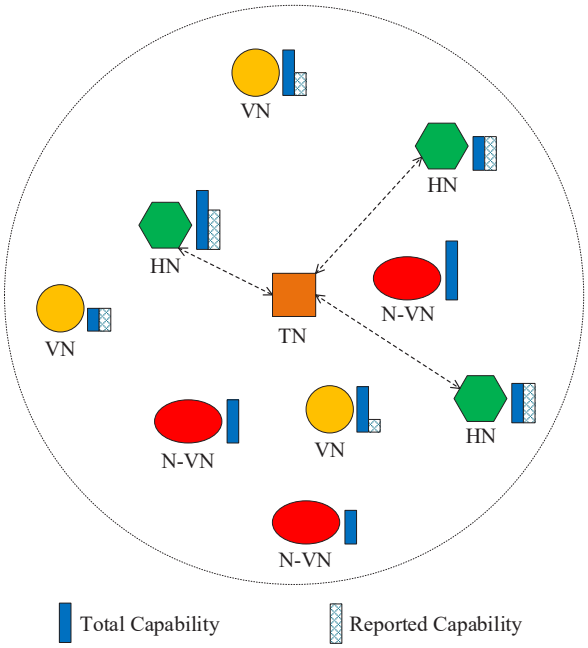


Fig. 1. Task scheduling service model among VNs in a fog network. An FN serves as a VN or an N-VN according to whether the FN is voluntary to contribute its capability to the TN. The VNs that are selected for the offloading service are defined as HNs. Obviously we have $\mathcal{H} \subseteq \mathcal{V} \subseteq \mathcal{F}$.

TABLE I
SUMMARY OF KEY NOTATIONS

Nota.	Unit	Description
l	bit	Overall task size of the TN.
$p_{\max}$	W	Upper bound of the transmission power of the TN.
p_i	W	Transmission power of the TN to FN i.
l_{T}	bit	Subtask size processed locally at the TN.
l_i	bit	Subtask size offloaded to FN i.
η_{T}	cycle/bit	CPU cycles for processing 1 bit data at the TN.
η_i	cycle/bit	CPU cycles for processing 1 bit data at FN i.
f_{T}	cycle/s	CPU frequency of the TN.
f_i	cycle/s	CPU frequency of FN i.
θ_i	J/cycle	Energy consumption per CPU cycle of FN i.
W	Hz	Spectrum bandwidth for task offloading.
γ_i	–	Path loss factor between the TN and FN i.
β_i	–	Shadowing factor between the TN and FN i.
τ_i	–	Capability report ratio of FN i.
κ_i	–	Regulatory factor of the report ratio of FN i.
E_i	J	Energy consumption of FN i in the offloading service of a task.
$\bar{E}_i$	J	Historical average energy consumption of FN i.
ω	–	Forgetting factor in the updating formula of the historical average FN energy consumption.
$\bar{E}_{i,\max}$	J	Threshold for the historical average energy consumption of FN i.

A. Task Scheduling among VNs

Once an l-bit delay-sensitive computing task is generated at the TN, in addition to processing the task locally or calling for cloud-assisted computing, the TN can call for task scheduling service from the FNs in the fog network, and then the FNs will report their information autonomously to the TN. According to the known FN and task information, the offloading solution is determined by the TN itself rather than a central controller. Then, the task is divided into subtasks and transmitted to the corresponding FNs for processing. Finally, processing results are transmitted back from the FNs to the TN if necessary. Through this kind of flexible fog-enabled task scheduling service, idle resources around the TN are utilized to achieve low task processing delay.

A report ratio τ_i is defined for each FN, which represents the proportion of the computing capability reported by FN i with respect to its total capability. The report ratio set is defined as $\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_N\}$. The total computing capability of FN i is $\frac{f_i}{\eta_i}$ (in bit/s), and the capability reported to the TN is

$$c_i = \tau_i \frac{f_i}{\eta_i}. \tag{1}$$

The report ratio satisfies $0 \leq \tau_i \leq 1$. If the report ratio of an FN is larger than 0, this FN is voluntary to contribute its capability for the task scheduling service, and it serves as a VN. Otherwise, the FN will not participate in the task scheduling service, and then it serves as an N-VN to the TN. The set of the VNs is defined as $\mathcal{V}$, which is a subset of $\mathcal{F}$ and contains the V FNs that report non-zero capabilities to the TN. It is worth mentioning that the VN set $\mathcal{V}$ may be time-varying according to the states of the FNs.

The VNs that are selected by the TN for task offloading are defined as HNs. Obviously, the sizes of the subtasks corresponding to the unselected VNs equal 0. Thus, the HN set $\mathcal{H}$ is a subset of $\mathcal{V}$, and it contains the VNs which are assigned subtasks with nonzero size. Therefore, we have $\mathcal{H} \subseteq \mathcal{V} \subseteq \mathcal{F}$.

B. Task Delay

According to the offloading process described above, the task delay includes $V+1$ parts. Specifically, they are the local processing delay of the subtask with l_{T} bits, and the delays of the subtasks (some of the subtasks may be 0) offloaded to the V VNs, which are denoted by d_{T} and $d_i (i \in \mathcal{V})$, respectively. In most fog network applications like image recognition, automatic control, and etc, the TN can make next-step decision only if the processing results of all the subtasks are received. Thus, the overall task delay is defined as the maximum value of all the subtask delays, i.e.

$$d = \max\left(d_{\mathrm{T}}, d_1, d_2, \cdots, d_V\right). \tag{2}$$

For the VN, say VN j, that is not selected as an HN for the offloading service, there is $l_j = 0$, and we define $d_j = \min(d_i), i \in \mathcal{H}$ for the sake of the later derivation. Thus, the definition of d in formula (2) still holds.

Following the models in [9], [26], the l_{T}-bit subtask processed locally at the TN requires $l_{\mathrm{T}}\eta_{\mathrm{T}}$ CPU cycles to be

computed, thus the local processing delay is expressed as

$$d_{\mathrm{T}} = \frac{l_{\mathrm{T}}\eta_{\mathrm{T}}}{f_{\mathrm{T}}}. \tag{3}$$

For the VN, say VN i, that is selected for the offloading service, the subtask size l_i is larger than 0, and the l_i-bit subtask requires $l_i\eta_i$ CPU cycles to be computed. This subtask needs to be transmitted to and processed at VN i, and the processing result is then transmitted back to the TN. In most cases, the processing result is a small packet such as a control signal [9]. Similar to the existing researches in [9], [28], [29], the transmission time of the processing result is neglected. Then, the offloading delay d_i is the sum of the transmission time and the processing time of the l_i-bit subtask, and it is expressed as

$$d_i = \frac{l_i}{WB_i} + \frac{l_i}{c_i} = l_i\left(\frac{1}{WB_i} + \frac{\eta_i}{\tau_i f_i}\right), \tag{4}$$

where B_i is the spectral efficiency of the wireless link from the TN to VN i. Given the TN transmission power p_i, B_i is obtained through Shannon capacity as

$$B_i = \log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right), \tag{5}$$

where γ_i and β_i are the path loss and shadowing factors of this wireless link. In addition, I_i and n_0 are the interference power and the noise power spectral density respectively.

C. Problem Formulation

Taking the fact that $l = l_{\mathrm{T}} + \sum_{i=1}^{V} l_i$ into consideration, the overall task processing delay in (2) can be rewritten as

$$d = \max_{i\in\mathcal{V}}\left[\frac{\left(l - \sum_{i=1}^{N} l_i\right)\eta_{\mathrm{T}}}{f_{\mathrm{T}}}, l_i\left(\frac{1}{W\log_2\left(1+\frac{p_i\gamma_i\beta_i}{I_i+Wn_0}\right)} + \frac{\eta_i}{\tau_i f_i}\right)\right]. \tag{6}$$

It can be observed that the overall task delay d is directly determined by the subtask size l_i and the transmission power p_i. Under the constraint for the overall transmission power of the TN, l_i and p_i need to be properly assigned thus the overall task processing delay is minimized. Now, the delay-optimal task scheduling problem is formulated as

$$\begin{aligned} \mathbf{P1}: \quad & \min_{\mathcal{L},\mathcal{P}} \quad d \\ & s.t. \quad 0 \le l_i \le l, \\ & \quad\;\; 0 \le p_i \le p_{\max}, \\ & \quad\;\; \sum_{i=1}^{V} p_i \le p_{\max}, \end{aligned} \tag{7}$$

where $p_{\max}$ is the upper bound of the overall transmission power of the TN. The sets $\mathcal{L}$ and $\mathcal{P}$ represent the subtask size set $\{l_1, l_2, \cdots, l_V\}$ and TN transmission power set $\{p_1, p_2, \cdots, p_V\}$ respectively.

Remark: According to the traffic load, energy consumption, etc., the capability report ratio τ_i in $\mathcal{T}$ is determined by the corresponding FN in a distributed and autonomous manner. Thus, the computing capability of FN i is $\tau_i\frac{f_i}{\eta_i}$ from the perspective of the TN. The report ratio $\tau_i > 0$ means that the corresponding FN is available for task scheduling service at present and serves as a VN. Given the reported capabilities of all the VNs, the optimal offloading solution $(\mathcal{L}^*, \mathcal{P}^*)$ that minimizes the task delay d is obtained at the TN by solving **P1**. The optimal HN set is $\mathcal{H}^* = \{\text{VN } i \in \mathcal{V} | l_i^* > 0\}$. The VNs with $l_i^* = 0$ and $p_i^* = 0$ are not selected as HNs for the delay-optimal task scheduling service, which may result from the low reported capabilities or the poor channel states of these unselected VNs.

In the next section, the capability report ratio of the FNs in the voluntary-mode fog networks is detailedly discussed and modeled.

III. Capability Report Ratio

The TN selects the HNs and offloads subtasks to the HNs according to the computing capability voluntarily reported by each FN. In this section, the capability report ratio is modeled, taking the historical energy consumption of the FN into consideration. Thus, a balanced offloading among the FNs can be achieved in the voluntary-mode fog networks.

A. Historical Energy Consumption

The FNs report their computing capabilities according to their historical energy consumptions. Therefore, we first introduce the update formula for the historical average energy consumption of FN i as

$$\bar{E_i}' = (1-\omega)\,\bar{E}_i + \omega E_i, \tag{8}$$

which is inspired by the proportional fair scheduling policy for the radio resource allocation [30], [31]. The parameter ω is the forgetting factor with a positive value less than 1. E_i is the energy consumption of FN i in the offloading service process of the TN task, and it is expressed as

$$E_i = l_i\eta_i\theta_i, \tag{9}$$

where θ_i is the energy consumption per CPU cycle of FN i. The energy consumption for the transmission of processing result is neglected like what we did in the formulation of task processing delay.

The historical energy consumption $\bar{E}_i$ reflects the energy consumption level of FN i in the past period. The FN with a higher historical energy consumption is able to autonomously reduce the capability reported to the TN, such that lighter load is assigned to this FN afterwards. Thus, balanced offloading and fair energy consumptions could be achieved among the FNs in the voluntary-mode fog networks.

B. Report Ratio Modeling

Based on the analyses above, the capability report ratio of FN i is modeled as the following function.

$$\tau_i = f\left(\bar{E}_i, \bar{E}_{i,\max}, \kappa_i\right) = \begin{cases} 1, & \bar{E}_i < \kappa_i \bar{E}_{i,\max}, \\ \dfrac{1}{1-\kappa_i}\left(1 - \dfrac{\bar{E}_i}{\bar{E}_{i,\max}}\right), & \kappa_i \bar{E}_{i,\max} \le \bar{E}_i \le \bar{E}_{i,\max}, \\ 0, & \bar{E}_i > \bar{E}_{i,\max}. \end{cases} \quad (10)$$

In formula (10), $\bar{E}_{i,\max}$ is the threshold of the historical energy consumption of FN i. If $\bar{E}_i$ is lower than the threshold, the report ratio τ_i of FN i is positive, which means that this FN is voluntary to contribute its capability and serves as a VN. The parameter κ_i with a positive value no larger than 1 is the regulatory factor of the report ratio. An illustration of the modeling for the report ratio is presented in Fig. 2.

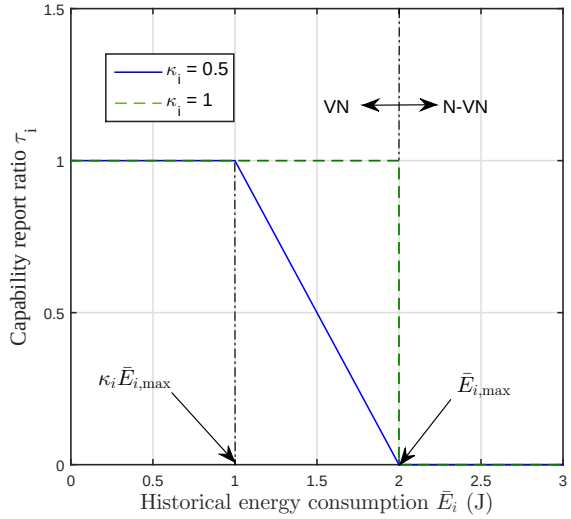


Fig. 2. Report ratio model of FN i.

As illustrated in Fig. 2 and formula (10), the report ratio τ_i is 1 when $\bar{E}_i \le \kappa_i \bar{E}_{i,\max}$, and all the computing capability is voluntarily reported to the TN that requires service. When $\bar{E}_i$ becomes larger than $\kappa_i \bar{E}_{i,\max}$, the report ratio becomes smaller than 1, which means FN i only reports part of its computing capability, i.e. $\tau_i \frac{f_i}{\eta_i}$, to the TN. It is equivalent to decreasing the CPU frequency of FN i to $\tau_i f_i$. When $\bar{E}_i$ becomes larger than the threshold $\bar{E}_{i,\max}$, a zero capability will be reported to the TN. FN i will not participate in the offloading service as the result of its high historical energy consumption. Thus, the historical average energy consumptions of the FNs are restricted by their thresholds of energy consumption, and a balanced offloading can be achieved with this capability report model. As shown in Fig. 2, when the regulatory factor κ_i equals 1, the corresponding FN will either report all of its capability or none of its capability to the TN.

With the report ratio, an FN in the fog networks serves as either a VN or an N-VN according to its historical energy consumption. Besides, the threshold $\bar{E}_{i,\max}$ and the regulatory factor κ_i is determined by the characteristics of the corresponding FN. Therefore, the reported capabilities are determined by the FNs in a distributed and autonomous manner. Small $\bar{E}_{i,\max}$ and κ_i indicate that the corresponding FN is sensitive to energy consumption. Obviously, if the thresholds of all the FNs are $+\infty$, all the computing capabilities of the FNs will be selflessly reported to the TN to fulfill the task offloading service, and the system will be in command-mode without without concerning the balanced offloading among the FNs.

Through the autonomous contribution of computing capability, this voluntary-mode task scheduling guarantees that the energy consumptions of individual FNs are not too large, and thus the balanced offloading is achieved. We adopt the Jain's fairness index [32] to numerically evaluate the fairness level of the energy consumptions among all the N FNs, and the fairness index is formulated as

$$F = \frac{\left(\sum_{i\in\mathcal{F}} \bar{E}_i\right)^2}{N \cdot \sum_{i\in\mathcal{F}} \bar{E}_i{}^2}. \quad (11)$$

The Jain's fairness index F ranges from $1/N$ to 1, and a higher F indicates a higher fairness level of the task scheduling scheme. The fairness index achieves the maximum value when all the FNs have equal historical energy consumptions.

We concentrate on the balanced offloading in the voluntary-mode fog networks in this paper, thus the report ratio is determined by the historical energy consumption of the corresponding FN. In the economic-mode fog networks, the TNs will provide the FNs with rewards as an incentive, by which the report ratios of the FNs are further determined.

In the next section, we are going to provide the delay-optimal task scheduling solution $(\mathcal{L}^*, \mathcal{P}^*)$ with the voluntary capability report in fog networks.

IV. Delay-Optimal Task Scheduling

In this section, the optimization problem $\mathbf{P1}$ is transformed into a one-variable form problem with respect to the terminal transmission power $\mathcal{P}$. Then, the Delay-Optimal Task Scheduling (DOTS) algorithm is proposed for the TN to obtain the solution of the delay-optimal problem.

A. Problem Transformation

As defined in formula (2), the overall task processing delay is determined by the maximum delay of all the $V+1$ subtasks. Then, we have the following proposition for the relationship between the overall task delay and the subtasks' delays.

Proposition 1. *When the overall task processing delay d is minimized, all the subtasks' delays are equal. In other words, we have*

$$d^* = d_{\mathrm{T}}^* = d_1^* = d_2^* = \cdots = d_V^*. \quad (12)$$

Proof. Please refer to Appendix A. ■

The conclusion in Proposition 1 can be intuitively explained as follows. If the subtasks' delays are not all equal, a lower overall task processing delay can always be achieved by adjusting the subtask sizes allocated to the network nodes.

Correlation between $\mathcal{L}$ and $\mathcal{P}$ is introduced by the conclusion in Proposition 1. Thus, the optimization problem $\mathbf{P1}$ can be transformed into a one-variable form problem provided in the following proposition.

Proposition 2. *The optimization problem* $\mathbf{P1}$ *is equivalent to the optimization problem with respect to the TN transmission power* $\mathcal{P}$*, which is formulated as*

$$\begin{aligned}\mathbf{P2}: \max_{\mathcal{P}} \ & K=\sum_{i=1}^{V}\left[\frac{1}{W\log_2\left(1+\frac{p_i\gamma_i\beta_i}{I_i+Wn_0}\right)}+\frac{\eta_i}{\tau_i f_i}\right]^{-1}+\frac{f_{\mathrm{T}}}{\eta_{\mathrm{T}}}\\ s.t. \ & 0\le p_i\le p_{\max},\\ & \sum_{i=1}^{V} p_i\le p_{\max}.\end{aligned} \tag{13}$$

Proof. Please refer to Appendix B. ■

Remark: From Appendix B, there is $l=d\cdot K$. The objective function K in $\mathbf{P2}$ is the equivalent processing rate of the offloading service based on the V VNs, which involves both the computing rates and the transmission rates. The overall task delay d is minimized when the maximum equivalent task processing rate K is achieved. Therefore, the delay minimization problem $\mathbf{P1}$ is transformed into a rate maximization problem $\mathbf{P2}$ with respect to $\mathcal{P}$.

Moreover, it can be found from (13) that the maximization of K is independent of the overall task size l. Therefore, the minimized task processing delay d^* holds a proportionable relationship with the TN task size l, i.e. $d^*=\frac{1}{K^*}\cdot l$, when the VN set $\mathcal{V}$ is fixed.

B. Optimal Offloading Solution

In this subsection, we propose the optimal TN transmission power $\mathcal{P}^*$ to maximize the equivalent task processing rate K in $\mathbf{P2}$. Then, the optimal offloading solution $(\mathcal{L}^*,\mathcal{P}^*)$ is provided to minimize the overall task processing delay.

Firstly, the convexity of K with respect to $\mathcal{P}$ is confirmed by the following proposition.

Proposition 3. *The equivalent processing rate of the offloading service, i.e.* K *in* $\mathbf{P2}$*, is convex when* $p_i\ge 0, i\in\mathcal{V}$*.*

Proof. Please refer to Appendix C. ■

Thus, the maximized objective function, i.e. K^*, can be obtained through convex optimization, which directly provides the minimized task processing delay as $\frac{l}{K^*}$. Then, we have the following theorem for the delay-optimal task scheduling solution $(\mathcal{L}^*,\mathcal{P}^*)$.

Theorem 1. *The optimal TN transmission power* $\mathcal{P}^*$ *that minimizes the overall task processing delay* d *satisfies the following conditions.*

$$p_i^*=\begin{cases}0, & \left.\frac{\partial K}{\partial p_i}\right|_{p_i=0}\le\alpha\\ \bar{p}_i, & \left.\frac{\partial K}{\partial p_i}\right|_{p_i=\bar{p}_i}=\alpha\\ p_{\max}, & \left.\frac{\partial K}{\partial p_i}\right|_{p_i=p_{\max}}\ge\alpha\end{cases} \tag{14}$$

$$\sum_{i=1}^{N} p_i=p_{\max}. \tag{15}$$

In the above equations, α *is a positive optimization parameter which is equal for all* V *VNs.* $\frac{\partial K}{\partial p_i}$ *is the first derivative of* K *with respect to* p_i*, and it is expressed as*

$$\begin{aligned}\frac{\partial K}{\partial p_i}=&W\left[1+W\frac{\eta_i}{\tau_i f_i}\log_2\left(1+\frac{p_i\gamma_i\beta_i}{I_i+Wn_0}\right)\right]^{-2}\\ &\cdot\frac{\gamma_i\beta_i}{(I_i+Wn_0+p_i\gamma_i\beta_i)\ln 2}.\end{aligned} \tag{16}$$

The optimal subtask size $\mathcal{L}^*$ *that minimizes the overall task delay is*

$$\mathcal{L}^*=\left\{\frac{l}{K|_{\mathcal{P}^*}}\left[\frac{1}{W\log_2\left(1+\frac{p_i^*\gamma_i\beta_i}{I_i+Wn_0}\right)}+\frac{\eta_i}{\tau_i f_i}\right]^{-1}, i\in\mathcal{V}\right\}. \tag{17}$$

Proof. Please refer to Appendix D. ■

Remark: The conclusions in Theorem 1 can be intuitionally explained as follows. Firstly, the processing rate K increases with the increasing of the transmission power $p_i, i\in\mathcal{V}$. Thus, the condition shown in (15) must be satisfied when K is maximized. Secondly, the gradient $\frac{\partial K}{\partial p_i}$ represents the increasing rate of K with respect to p_i, which is related to the computing capability and the channel state of the corresponding VN. The gradients $\frac{\partial K}{\partial p_i}, (i\in\mathcal{V})$ should be equal when K is maximized. Otherwise, $\mathcal{P}$ can always be adjusted to obtain a larger K. Thirdly, due to the constraint for TN transmission power, i.e. $p_{\max}$, the VN with very poor channel state or low computing capability may hold a very low $\frac{\partial K}{\partial p_i}$ and not be selected as an HN for the task scheduling service. On the contrary, the VN with very good channel state or high computing capability may hold a very high $\frac{\partial K}{\partial p_i}$ and be assigned all transmission power $p_{\max}$.

C. DOTS Algorithm

Based on the conclusions in Theorem 1, the DOTS algorithm is proposed in Algorithm 1.

Remark: In Algorithm 1, we adopt iterative method to obtain the optimal TN transmission power $\mathcal{P}^*$ provided in Theorem 1. A precision ε and an initial step size Δp are initialized. In addition, the initial transmission power to all V VNs are set to be equal, i.e. $\frac{p_{\max}}{V}$. Iterations are carried out to approach the optimal solution provided in Theorem 1. The step size could be set larger at the beginning, and be reduced in the process of iterations. Thus, a faster convergence of $\mathcal{P}$ could be achieved. The optimal subtask size $\mathcal{L}^*$ could be obtained

Algorithm 1 DOTS Algorithm

Initialize $\bar{E}_i$, $\bar{E}_{i,\max}$, κ_i $(i \in \mathcal{F})$, ω, Δp, ε;
while A TN task is generated **do**
  Acquire the report ratio set $\mathcal{T}$ with formula (10);
  Update the VN set $\mathcal{V} = \{\text{FN } i \in \mathcal{F} | \tau_i > 0\}$;
  Initialize $\mathcal{P} = \left\{ \frac{p_{\max}}{V}, \frac{p_{\max}}{V}, \cdots, \frac{p_{\max}}{V} \right\}$;
  Calculate $\frac{\partial K}{\partial \mathcal{P}} = \left\{ \frac{\partial K}{\partial p_1}, \frac{\partial K}{\partial p_2}, \cdots, \frac{\partial K}{\partial p_V} \right\}$;
  $s1 = \text{std}\left\{ \frac{\partial K}{\partial \mathcal{P}} | p_i \neq 0 \right\}$;
  while $s1 > \varepsilon$ **do**
    $\mathcal{P}_{\text{temp}} = \mathcal{P}$;
    Find $i = \arg\min \left\{ \frac{\partial K}{\partial \mathcal{P}} | p_i \neq 0 \right\}$;
    $p_i = \max(p_i - \Delta p, 0)$;
    Find $j = \arg\max \left\{ \frac{\partial K}{\partial \mathcal{P}} \right\}$;
    $p_j = \min(p_j + \Delta p, p_{\max})$;
    Update the gradient $\frac{\partial K}{\partial \mathcal{P}}$;
    $s2 = \text{std}\left\{ \frac{\partial K}{\partial \mathcal{P}} | p_i \neq 0 \right\}$;
    if $s2 > s1$ **then**
      Reduce the step size as $\Delta p = \frac{\Delta p}{2}$;
      $\mathcal{P} = \mathcal{P}_{\text{temp}}$;
      Update the gradient $\frac{\partial K}{\partial \mathcal{P}}$;
    else
      $s1 = s2$;
    end if
  end while
  return Optimal TN transmission power $\mathcal{P}^* = \mathcal{P}$;
      Optimal subtask sizes $\mathcal{L}^*$ calculated by (17);
      Optimal HN set $\mathcal{H}^* = \{\text{VN } i \in \mathcal{V} | l_i^* > 0\}$;
  Update the historical energy consumption for each FN with formula (8);
end while

through formula (17) in Theorem 1. The optimal HN set $\mathcal{H}^*$ contains the selected VNs, i.e. the VNs with optimal subtask sizes larger than 0.

V. Numerical Evaluations

In this section, simulations are carried out to investigate the performance of the proposed DOTS algorithm. The minimized task processing delay is evaluated in various scenarios. Besides, the FN energy consumptions and the fairness level among the FNs are also investigated.

A. Simulation Setting

A voluntary-mode fog network consisting of N FNs is considered for the numerical evaluations, and the FNs are randomly distributed in the fog cluster. The TN in this fog network calls for and receives offloading services from the VNs based on their voluntarily reported computing capabilities. The transmission bandwidth for each FN is 10 MHz. The interference power I_i, the noise power density n_0, and the shadowing factor β_i are -47 dBm, -173 dBm/Hz, and -5 dB, respectively. In addition, the path loss factor γ_i (in dB) is calculated by $38.46 + 20\log_{10}(D_i)$, where D_i (in m) is the distance between the TN and FN i. All parameter settings are provided in Table. II.

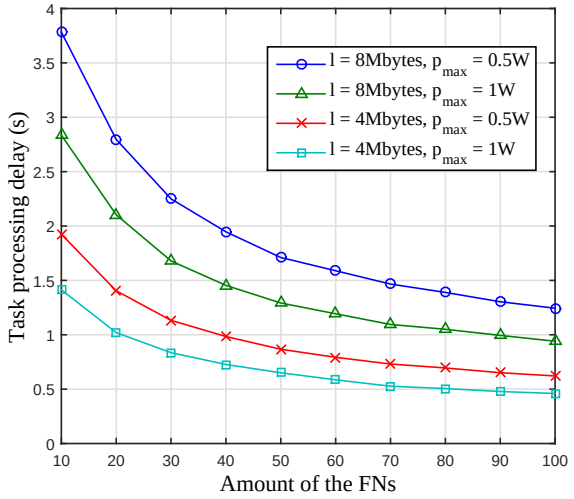
TABLE II
Simulation Parameters

Parameter	Value
The radius of the fog cluster	$10 - 90$ m
W	10 MHz
N	$10 - 100$
l	$[4, 6, 8]$ MBytes
$p_{\max}$	$[0.5, 1]$ W
η_{T}	1000 cycle/bit
η_i	$500 - 1500$ cycle/bit
f_{T}	2 GHz (cycle/s)
f_i	$1 - 10$ GHz (cycle/s)
θ_i	$[1 - 10] \times 10^{-10}$ J/cycle
κ_i	$0 - 1$
$\bar{E}_{i,\max}$	$[0.5 - \infty)$ J
ω	0.01

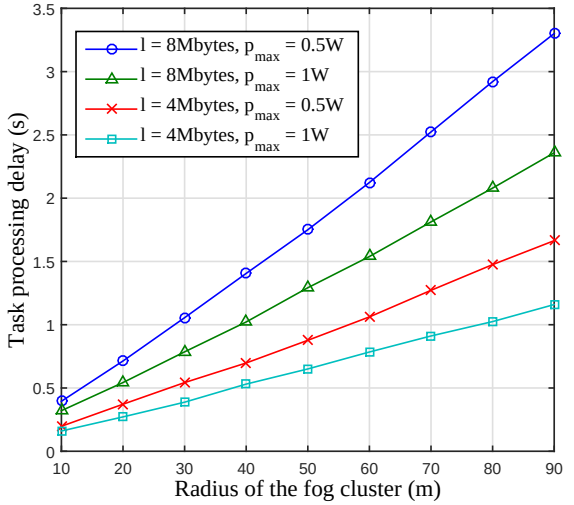
B. Minimized Task Delay

Given the reported computing capabilities in the fog cluster, the optimal offloading solution that minimizes the overall task processing delay is obtained by the proposed DOTS algorithm.

In Fig. 3, the minimized task processing delay, i.e. d^*, is provided with different task sizes l and transmission power bounds $p_{\max}$. In Fig. 3a, the change of the minimized task delay over the total FN amount in the fog cluster is plotted, wherein the radius of the fog cluster is 50 m. With the increasing of the total FN amount, the amount of the VNs that are close to the TN and report high computing capability increases, which brings an increasing task processing rate K and thus a decreasing task processing delay d. Besides, the higher upper bound for TN transmission power also leads to lower task processing delays by increasing the transmission rate in K. In addition, it can also be observed from the results in this figure that the decreasing rate of the task delay becomes smaller and smaller with the increasing of the total FN amount. This is because that the resources in the fog cluster that the TN can utilize are limited by its transmission power, and the TN can only makes use of finite VNs for the offloading of its computing tasks. In other words, the offloading processes of the delay-sensitive TN tasks are transmission power-constrained when the total FN amount is large. In Fig. 3b, the change of the minimized task delay over the radius of the fog cluster is plotted, wherein the total FN amount in the cluster is 50. With the increasing of the radius of the fog cluster, the amount of VNs that are close to the TN and report high computing capability decreases, which brings a decreasing task processing rate K and thus an increasing task processing delay d. In addition, it can be observed that the increasing of the task delay is nearly linear with the increasing of the cluster radius, which also indicates that the resources that the TN can utilize with limited transmission power are proportional with the radius of the fog cluster.



(a) The overall task delay versus the FN amount in the fog cluster.



(b) The overall task delay versus the radius of the fog cluster.

Fig. 3. The minimized overall task processing delay under different task sizes and transmission power bounds.

Moreover, a nearly proportional relationship between the minimized task delay and the task size is found from the results in Fig. 3. This has been explained in the remark of Proposition 1, and the deviations come from the varying of the VN set. The minimization of the task delay is equivalent to the maximization of the total processing rate K, and the equivalent total processing rate K is independent of the task size l. In other words, the tasks with different sizes have the equal maximal processing rate. Therefore, the minimized delay $d^* = \frac{1}{K^*} \cdot l$ is linear with the task size l.

In Fig. 4, the change of the average task processing delay over the threshold for FN energy consumption is plotted, wherein the regulatory factor is 0.5 for each FN. With the increasing of the threshold, the VNs report more capabilities, which brings a decreasing average task processing delay. Besides, the task processing delay is almost unchanged when the thresholds are large enough, which results from two reasons.

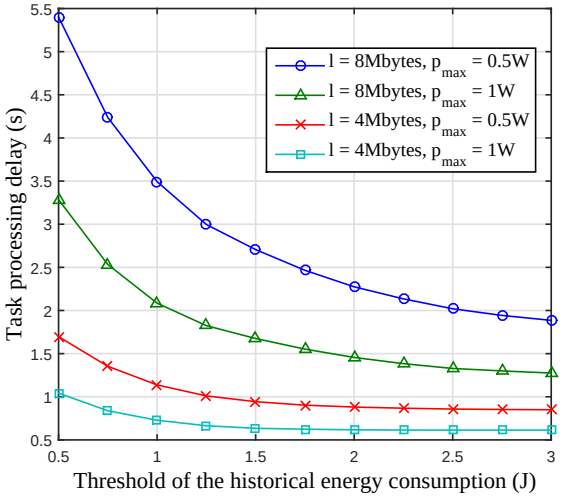


Fig. 4. The average task processing delay versus the threshold for FN energy consumption.

Firstly, when the thresholds are large, the resources that the TN can utilize is not limited by the reported capabilities of the nearby VNs, but the upper bound of the TN transmission power. Secondly, when the thresholds are large enough, all FNs in the fog network report all their computing capabilities to the TN, and the offloading services are in command-mode.

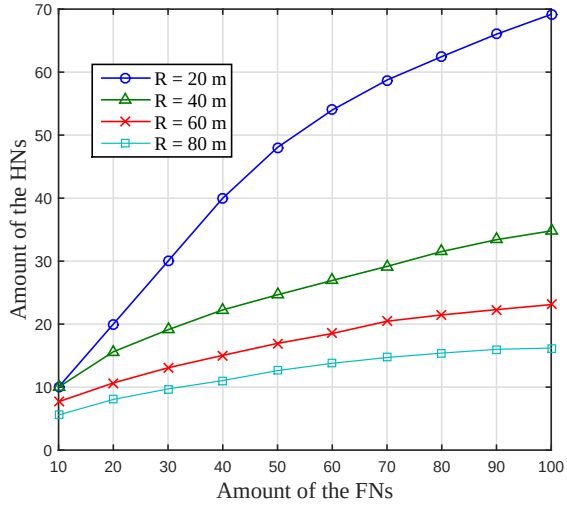


Fig. 5. The amount of the HNs versus the amount of the FNs.

In the optimal offloading solution $(\mathcal{L}^*, \mathcal{P}^*)$, the subtask size l_i^* may be zero which means VN i is not selected as an HN. In Fig. 5, the amount of the HNs is provided with varying total FN amount and cluster radius, wherein the task size $l = 6$ MBytes and the upper bound for TN transmission power $p_{\max} = 1$ W. It can be observed that the amount of the selected FNs increases with the increasing of the total FN amount and the decreasing of the fog cluster radius, and the increasing rate becomes smaller when the total FN amount becomes larger. Taking the results in Fig. 3 and Fig. 5 into consideration,

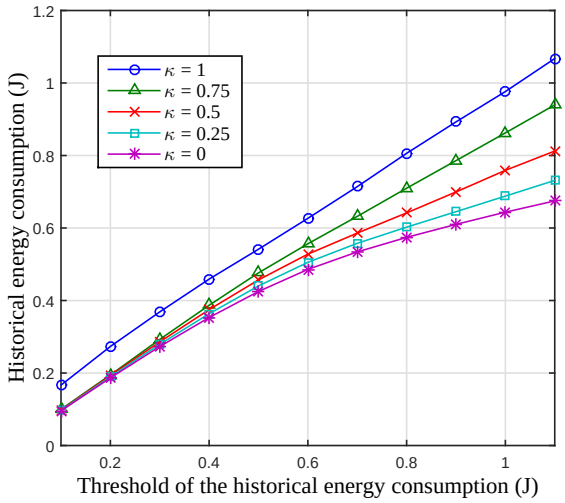
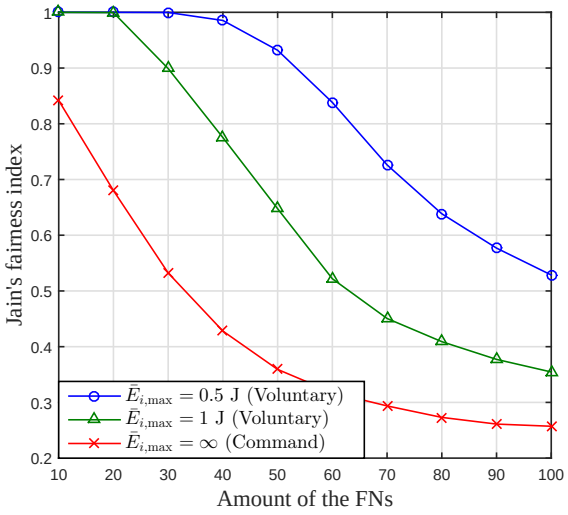
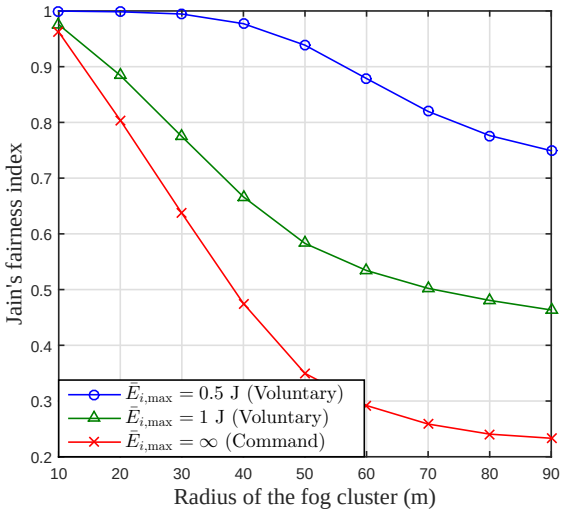


Fig. 6. The historical FN energy consumption versus the threshold for FN energy consumption.



(a) Jain's fairness index versus the FN amount in the fog cluster.



(b) Jain's fairness index versus the radius of the fog cluster.

Fig. 7. Jain's fairness index under different thresholds for FN energy consumption.

we can conclude that the appropriate amount of FNs and a small distribution range are advantageous for the offloading of the delay-sensitive TN tasks. Overmany deployed FNs can not dramatically reduce the task processing delay due to the limited TN transmission power, but result in a high networking overhead.

C. Fairness Index

For a series of TN tasks, the voluntary capability report supported by the distributed report ratio guarantees that the energy consumptions of the FNs are not too high to end their lifecycles quickly. In Fig. 6, the historical average energy consumption of an FN is plotted under different threshold $E_{i,\max}$ and regulatory factor κ_i, wherein the task size is 6 Mbytes. It can be observed from the results that when the regulatory factor is smaller than 1, the historical FN energy consumption can always be effectively kept lower than the threshold, and the historical energy consumption will break the threshold when $\kappa_i = 1$. Therefore, a positive regulatory factor smaller than 1 is preferred in our capability report ratio model, and the transition zone of the report ratio ($\kappa_i = 0.5$) shown in Fig. 2 is necessary.

The Jain's fairness indexes F of the FN energy consumption under different thresholds for FN energy consumption are plotted in Fig. 7, wherein the task size is 6 Mbytes and the TN transmission power bound is 1 W. Besides, the regulatory factor κ is set to be 0.5. In Fig. 7a, the changes of the Jain's fairness index over the total amount of the FNs are plotted, wherein the cluster radius is 50 m. In Fig. 7b, the changes of the Jain's fairness index over the radius of the fog cluster are plotted, wherein the total FN amount is 50.

The simulation results with infinite threshold for FN energy consumption reflect the fairness level of command-mode offloading. When the system is in command-mode, all the computing capabilities of the FNs are reported to the TN to achieve the minimized task processing delay, regardless of the energy consumptions of the FNs. As a result, the fairness level of this command-mode offloading is much lower than that of the voluntary-mode offloading with finite threshold for FN energy consumption. Besides, a lower threshold leads to a higher fairness level. It is because that when the threshold for FN energy consumption is small, the FNs with higher historical load will report less or zero capability, and more FNs participate in the offloading services of the TN tasks.

Moreover, the fairness level decreases with the increasing of the FN amount and the cluster radius as shown in the simulation results. The reason is that the amount of the FNs which are far away from the TN increases, and these FNs can hardly be adopted for the task scheduling. Therefore, except for the benefit to reducing task processing delay, the appropriate amount and distribution range of the FNs in the fog cluster also lead to higher fairness level among the FNs during the offloading services for TN tasks.

VI. Conclusions

In order to cope with the requirement for ultra-low processing delay in future 5G and IoT applications, the novel fog architecture with FNs located across the network has arisen to support nearby offloading services. In this paper, we investigated the scheduling of delay-sensitive tasks in the voluntary-mode fog networks, wherein the FNs voluntarily contribute their capabilities to the TNs and serves as VNs according to a distributed capability report ratio. An efficient DOTS algorithm was proposed to obtain the delay-optimal task scheduling solution. Numerical results obtained from extensive simulations confirmed that the DOTS algorithm can effectively obtain the optimal offloading solution to minimize the overall task processing delay. Moreover, balanced energy consumptions and higher fairness level among the FNs were achieved in the voluntary-mode fog networks based on the proposed voluntary capability report ratio. Future work is in progress to consider the incentive and bidding mechanisms in the voluntary-mode and economy-mode fog networks.

Appendix A
Proof of Proposition 1

We prove Proposition 1 with contradiction. In other words, if the condition that $d = d_{\mathrm{T}} = d_1 = d_2 = \cdots = d_V$ is not satisfied, we can always achieve $d' < d = \max(d_{\mathrm{T}}, d_1, d_2, \cdots, d_V)$.

Assume there is a offloading solution $(\mathcal{L}, \mathcal{P})$, and the corresponding delays do not satisfy the condition that $d = d_{\mathrm{T}} = d_1 = d_2 = \cdots = d_V$. Then, we can always obtain

$$a = \arg\min\left(d_{\mathrm{T}}, d_1, d_2, \cdots, d_V\right), \tag{18a}$$

$$b = \arg\max\left(d_{\mathrm{T}}, d_1, d_2, \cdots, d_V\right), \tag{18b}$$

which satisfy $l_b, l_a > 0$. Because that we previously define $d_j = \min_{l_i \neq 0}(d_i), i \in \mathcal{V}$ when $l_j = 0$. Obviously, there is $d_a < d_b = d$. Then, we can always get a small subtask adjustment parameter $\xi > 0$, such that

$$d'_a|_{l'_a = l_a + \xi} < d_b, \tag{19a}$$

$$d'_b|_{l'_b = l_b - \xi} < d_b. \tag{19b}$$

Thus, the overall task delay d' with adjusted offloading solution $(\mathcal{L}', \mathcal{P})$ is lower than d. That is to say, we can always achieve a lower overall delay by adjusting the subtask sizes if the condition in Proposition 1 is not satisfied.

The above proves Proposition 1.

Appendix B
Proof of Proposition 2

According to the conclusion in Proposition 1, we have

$$d_{\mathrm{T}} = d = \frac{\left(l - \sum_{i=1}^{V} l_i\right)\eta_{\mathrm{T}}}{f_{\mathrm{T}}}, \tag{20}$$

$$l_i = d\left[\frac{1}{W\log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right)} + \frac{\eta_i}{\tau_i f_i}\right]^{-1}, i \in \mathcal{V}. \tag{21}$$

Substituting (21) into (20) gives

$$\frac{l}{d} = \sum_{i=1}^{V}\left[\frac{1}{W\log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right)} + \frac{\eta_i}{\tau_i f_i}\right]^{-1} + \frac{f_{\mathrm{T}}}{\eta_{\mathrm{T}}}. \tag{22}$$

Denote the right part in (22) as K, then we get

$$\min_{\mathcal{P}} d \;\Leftrightarrow\; \max_{\mathcal{P}} \frac{l}{d} \;\Leftrightarrow\; \max_{\mathcal{P}} K. \tag{23}$$

Therefore, the minimization of the overall task delay d is transformed into the maximization of K with respect to $\mathcal{P}$. The above proves Proposition 2.

Appendix C
Proof of Proposition 3

The first derivative of K with respect to p_i is

$$\frac{\partial K}{\partial p_i} = W\left[1 + W\frac{\eta_i}{\tau_i f_i}\log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right)\right]^{-2} \cdot \frac{\gamma_i\beta_i}{(I_i + Wn_0 + p_i\gamma_i\beta_i)\ln 2}. \tag{24}$$

Then, the second derivative of K with respect to p_i is

$$\frac{\partial^2 K}{\partial p_i^2} = -W\left[\frac{\gamma_i\beta_i}{(I_i + Wn_0 + p_i\gamma_i\beta_i)\ln 2}\right]^2 \cdot \left\{2W\frac{\eta_i}{\tau_i f_i}\left[1 + W\frac{\eta_i}{\tau_i f_i}\log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right)\right]^{-3} + \ln 2\left[1 + W\frac{\eta_i}{\tau_i f_i}\log_2\left(1 + \frac{p_i\gamma_i\beta_i}{I_i + Wn_0}\right)\right]^{-2}\right\}, \tag{25}$$

which is strictly less than 0 when $p_i \geq 0$. Thus, the objective function K in $\mathbf{P2}$ is convex. The above proves Proposition 3.

Appendix D
Proof of Theorem 1

According to the convexity of K, the constrained optimization problem $\mathbf{P2}$ can be transformed into an unconstrained optimization problem as

$$L(\mathcal{P}, \alpha, \Psi, \Phi) = K(\mathcal{P}) + \alpha\left(p_{\max} - \sum_{i=1}^{V} p_i\right) + \sum_{i=1}^{V}\psi_i p_i + \sum_{i=1}^{V}\phi_i(p_{\max} - p_i), \tag{26}$$

where $\mathcal{P} = \{p_1, p2, \cdots, p_V\}$, $\Psi = \{\psi_1, \psi_2, \cdots, \psi_V\}$, and $\Phi = \{\phi_1, \phi_2, \cdots, \phi_V\}$.

In order to obtain the optimal TN transmission power, the Karush-Kuhn-Tucker (KKT) conditions [33] can be expressed as

$$\frac{\partial L}{\partial p_i} = \frac{\partial K}{\partial p_i} - \alpha + \psi_i - \phi_i = 0, \tag{27a}$$

$$\alpha\left(p_{\max} - \sum_{i=1}^{V} p_i\right) = 0, \tag{27b}$$

$$\psi_i p_i = 0, \tag{27c}$$

$$\phi_i(p_{\max} - p_i) = 0, \tag{27d}$$

$$0 \le p_i \le p_{\max}, \tag{27e}$$

$$\alpha, \psi_i, \phi_i \ge 0. \tag{27f}$$

If $p_i = 0$, then we have $\phi_i = 0$. Thus, we get

$$\frac{\partial K}{\partial p_i}\Big|_{p_i=0} - \alpha + \psi_i = 0 \xrightarrow{\psi_i \ge 0} \frac{\partial K}{\partial p_i}\Big|_{p_i=0} \le \alpha. \tag{28}$$

If $p_i = p_{\max}$, then we have $\psi_i = 0$. Thus, we get

$$\frac{\partial K}{\partial p_i}\Big|_{p_i=p_{\max}} - \alpha - \phi_i = 0 \xrightarrow{\phi_i \ge 0} \frac{\partial K}{\partial p_i}\Big|_{p_i=p_{\max}} \ge \alpha. \tag{29}$$

If $0 < p_i < p_{\max}$, then we have $\phi_i = 0$ and $\psi_i = 0$. Thus, we get

$$\frac{\partial K}{\partial p_i}\Big|_{p_i=\bar{p}_i} - \alpha = 0 \longrightarrow \frac{\partial K}{\partial p_i}\Big|_{p_i=\bar{p}_i} = \alpha. \tag{30}$$

In this case, the value of p_i is determined by the formula (30).

The value of $\frac{\partial K}{\partial p_i}$ is strictly positive according to (24). Thus, the value of α is positive when $0 \le p_i < p_{\max}$ according to (28) and (30), and thus $p_{\max} - \sum_{i=1}^{V} p_i = 0$ when $0 \le p_i < p_{\max}$ according to (27b). Besides, when $p_i = p_{\max}$, there must be another VN, say VN j, to which the TN transmission power is $p_j = 0$. In this case, the condition that $\alpha > 0$ and $p_{\max} - \sum_{i=1}^{V} p_i = 0$ are also satisfied.

According to Proposition 1, the subtask delay $d_i^* = d^*$ when d is minimized. Thus, we have

$$\begin{aligned} l_i^* =& d^* \left[\frac{1}{W \log_2\left(1 + \frac{p_i^* \gamma_i \beta_i}{I_i + W n_0}\right)} + \frac{\eta_i}{\tau_i f_i} \right]^{-1} \\ =& \frac{l}{K|_{\mathcal{P}^*}} \left[\frac{1}{W \log_2\left(1 + \frac{p_i^* \gamma_i \beta_i}{I_i + W n_0}\right)} + \frac{\eta_i}{\tau_i f_i} \right]^{-1}. \end{aligned} \tag{31}$$

The above proves Theorem 1.

REFERENCES

[1] G. Zhang, F. Shen, Y. Zhang, R. Yang, Y. Yang, and E. Jorswieck, "Delay minimized task scheduling in fog-enabled IoT networks," in *IEEE International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2018, pp. 1–6.

[2] Y. Yang, J. Xu, G. Shi, and C.-X. Wang, "5G wireless systems: Simulation and evaluation techniques," *Springer*, 2017.

[3] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Survey & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 3rd Quarter 2016.

[4] S. Li, L. D. Xu, and S. Zhao, "The Internet of Things: A Survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015.

[5] Cisco, "Visual networking index: Global mobile data traffic forecast update, 2015-2020," 2016.

[6] A. Botta, W. D. Donato, and V. Persico, "Integration of cloud computing and Internet of Things," *Future Generation Computer Systems*, vol. 56, no. C, pp. 684 – 700, Mar. 2016.

[7] M. Chiang, S. Ha, I. Chih-Lin, F. Risso, and T. Zhang, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 18–20, Apr. 2017.

[8] E. Ahmed, A. Naveed, A. Gani, S. H. A. Hamid, M. Imran, and M. Guizani, "Process state synchronization for mobility support in mobile cloud computing," in *2017 IEEE International Conference on Communications (ICC)*, May. 2017, pp. 1–6.

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

[10] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46 – 54, Jan. 2016.

[11] Y. J. Ku, D. Y. Lin, C. F. Lee, P. J. Hsieh, H. Y. Wei, C. T. Chou, and A. C. Pang, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 46–52, Apr. 2017.

[12] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[13] V. G. Nguyen, A. Brunstrom, K. J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1567–1602, 3rd Quarter 2017.

[14] N. Chen, Y. Yang, T. Zhang, M. Zhou, X. Luo, and J. K. Zao, "Fog as a service technology," *IEEE Communications Magazine*, pp. 1–7, 2018.

[15] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 416–464, 1st Quarter 2018.

[16] V. B. C. Souza, W. Ramłrez, X. Masip-Bruin, E. Marłn-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *2016 IEEE International Conference on Communications (ICC)*, May. 2016, pp. 1–5.

[17] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," in *2017 IEEE International Conference on Edge Computing (EDGE)*, Jun. 2017, pp. 17–24.

[18] S. Zhao, Y. Yang, Z. Shao, X. Yang, H. Qian, and C. X. Wang, "FEMOS: Fog-enabled multitier operations scheduling in dynamic wireless networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1169–1183, Apr. 2018.

[19] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, May. 2017.

[20] Y. Y. Shih, W. H. Chung, A. C. Pang, T. C. Chiu, and H. Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Network*, vol. 31, no. 1, pp. 52–58, Jan. 2017.

[21] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[22] 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 Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[23] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, and J. Wang, "DEBTS: Delay energy balanced task scheduling in homogeneous fog networks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2094–2106, Jun. 2018.

[24] Y. Sun and N. Zhang, "A resource-sharing model based on a repeated game in fog computing," *Saudi Journal of Biological Sciences*, vol. 24, no. 3, pp. 687–94, Mar. 2017.

[25] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, "A hierarchical game framework for resource management in fog computing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 52–57, Aug. 2017.

[26] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M.-T. Zhou, "MEETS: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4076–4087, Qct. 2018.

[27] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[28] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[29] C. Xian, Y.-H. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proceedings of the 13th International Conference on Parallel and Distributed Systems - Volume 01*, Dec. 2007, pp. 1–8.

[30] F. Afroz, K. Sandrasegaran, and P. Ghosal, "Performance analysis of PF, M-LWDF and EXP/PF packet scheduling algorithms in 3GPP LTE downlink," in *2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, Nov. 2014, pp. 87–92.

[31] G. Zhang, J. Xu, L. Liu, Y. Yang, Q. Li, and M. Hamalainen, "Theoretical analysis of PF scheduling with bursty traffic model in OFDMA

systems," in *IEEE International Conference on Communications (ICC)*, May. 2017, pp. 1–6.

[32] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *CoRR*, vol. cs.NI/9809099, Jan. 1998.

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