# Performance Analysis and Optimization of Delayed Offloading System With Opportunistic Fog Node

Haneul Ko, *Member, IEEE*, and Yeunwoong Kyung

***Abstract*—Fog node (FN) close to Internet of Things (IoT) devices can be exploited to offload the computing task of IoT devices. However, when lots of tasks are simultaneously offloaded from multiple IoT devices, FN can be overloaded. To mitigate this problem, we introduce a DYSON: DelaYed offloading System with Opportunistic fog Node (OFN) such as vehicles and mobile phones which can opportunistically reduce the load of static FN thanks to their mobility. In DYSON, to maximize the offloading effect of OFN, when IoT device cannot use OFN, it delays the task offloading with the expectation of future contacts of OFN. To assess the performance of DYSON, we develop an analytic model for the opportunistic offloading probability that the task can be offloaded to OFN. Based on the analytic model, we derive the optimal delay timer to maximize the opportunistic offloading probability while maintaining the probability that the task cannot be processed within the deadline below a target probability. Extensive simulation results are provided to show the superiority of DYSON.**

***Index Terms*—Delayed offloading, fog computing, load reduction, opportunistic fog, task offloading.**

## I. Introduction

In these days, along with the development of Internet of Things (IoT) applications, the cloud computing architecture has been used for the task processing of IoT devices owing to the flexibility and scalability on the resource management of the cloud service [1]. However, the physical distance between the cloud servers and IoT devices leads to long service delay and consumes lots of bandwidth and energy of the network. This triggers the concept of the fog computing where computing resources are located close to the IoT devices [2], [3].

In the fog computing architecture, tasks from IoT devices usually can be offloaded to the fog node (FN) which is connected to the access network device (e.g., a cellular base station (BS) [4]). However, even though FN is promising to offload the tasks, the scalability issue can occur because of limited computation resources and latency requirement of IoT applications [5]. For example, if lots of IoT devices exist at similar location and offload excessive tasks at the same time, FN can be easily overloaded. To solve this problem, there are several works (e.g., co-operation with other FNs and remote cloud) [4], [6]. One of promising solutions is to introduce opportunistic FNs (OFNs). OFN denotes FN with the mobility, such as vehicles and mobile phones, to which IoT devices can offload their tasks [7]. OFNs can connect with IoT devices and provide resources to them only when they are physically near IoT devices. That is, IoT devices opportunistically offload tasks to OFNs when OFNs are close to them [5], [8]–[12]. In [5], moving vehicles are utilized as OFNs to reduce the average system response time. Zhu *et al.* [8] provide a task offloading scheme to shorten the average service latency while reducing the overall quality loss when the nearby OFNs are discovered. In [9], predictive offloading with OFNs is introduced to distribute the load of static FN that suffers from its resource limitation and save the backhaul resources. Under information asymmetry and uncertainty in the vehicular fog computing (VFC) scenario, Zhou *et al.* [10] present a contract-based server recruitment method to share the available resources and propose matching-learning-based task offloading method that matches user vehicles and OFNs to minimize the delay performance. Meanwhile, privacy and security concerns of offloading also have been considered [11], [12]. In [11], blockchain is utilized to validate offloaded data transactions among OFNs and cloud nodes. In addition, Liao *et al.* [12] introduce the blockchain-based VFC framework where BS plays the consensus role in the blockchain network. However, in these works, tasks can be offloaded to OFN only when there is available OFN. That is, if there is no OFN, tasks have to be unwillingly handled by static FN.

To offload more tasks to OFN,[1] we introduce the DelaYed offloading System with OFN, named DYSON. In DYSON, the offloading of the task can be delayed with the expectation of future OFN contacts. That is, when IoT device cannot use OFN (i.e., there is no available OFN close to IoT device), it delays the offloading up to a pre-defined delay timer. Then, if OFN appears before the delay timer expires, the pending task can be delivered to OFN, which means higher utilization of OFN. Meanwhile, after the delay timer expires, IoT device always offloads its task to FN. To evaluate the performance of DYSON, we develop the analytical model for the opportunistic offloading probability that the task of IoT device is offloaded to OFN. Note that the opportunistic offloading probability can be considered as the load distribution effect. Based on the developed model, we derive the optimal delay timer to maximize the opportunistic offloading probability while maintaining the probability that the task cannot be processed within the deadline below a target probability. Extensive simulation results validate the analytic model and demonstrate that DYSON with the optimal timer has up to 298% opportunistic offloading probability with 1% outage probability compared to that of the conventional on-the-spot offloading system which offloads the task without the delay timer.

The main contribution is as follow: 1) even though the benefits of delayed offloading are considered in Wi-Fi offloading to reduce the traffic over the cellular networks [13]–[15], to the best of our knowledge, this is the first analytic work that considers the delayed task offloading in the fog computing architecture considering OFN. Specifically, we develop the analytic models for the opportunistic offloading probability and derive the optimal delay timer to maximize the offloading effect while considering the task completion deadline;[2] and 2) extensive

Manuscript received 18 June 2021; revised 9 February 2022 and 8 May 2022; accepted 30 May 2022. Date of publication 2 June 2022; date of current version 19 September 2022. This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant 2020R1G1A1100493. The review of this article was coordinated by Prof. K. Lee. *(Corresponding author: Yeunwoong Kyung.)*

Haneul Ko is with the Department of Computer and Information Science, Korea University, Sejong, South Korea (e-mail: heko@korea.ac.kr).

Yeunwoong Kyung is with the School of Computer Engineering, Hanshin University, Osan 18101, South Korea (e-mail: ywkyung@hs.ac.kr).

Digital Object Identifier 10.1109/TVT.2022.3179658

[1]Note that, when individual IoT devices offload tasks to OFN as much as possible, FN is not overloaded naturally even with multiple IoT devices.

[2]The derivation method for the optimal delay timer of the delayed task offloading is totally different from that of the delayed Wi-Fi offloading due to the difference of the offloading time point. Specifically, in the delayed Wi-Fi offloading, the agent can offload its data download through Wi-Fi AP in a continuous time space. On the other hand, in this paper, the agent offloads the task only at a specific time point to OFN (i.e., one-shot task offloading). In addition, in the mobile environment, the delayed task offloading is more practical than the delayed Wi-Fi offloading because each task processing procedure (i.e., offloading, processing, and response) for the delayed task offloading can be performed with an independent session (i.e., the session and service managements such as socket migration and service synchronization do not need to be considered in the delayed task offloading).

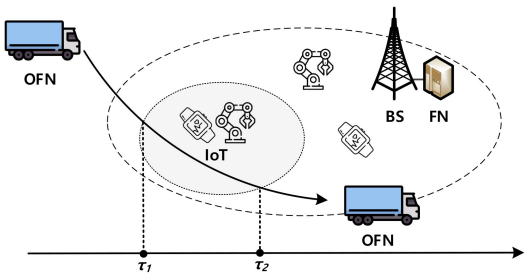


Fig. 1. System model.

simulation results demonstrate the performance of the conventional and proposed systems under various settings, which can be utilized to design the valuable guidelines for advanced OFN-based architectures.

The remainder of this paper is organized as follows. DYSON is introduced in Section II. Performance analysis and simulation results are provided in Sections III and IV, respectively. Finally, this paper is concluded with future works in Section V.

## II. Delayed Offloading System With Opportunistic Fog Nodes (DYSON)

Fig. 1 shows the system model of this paper. It is assumed that FN attached to BS is always available whereas OFN is only available when it moves close enough to IoT device. In Fig. 1, the grey circle centered on IoT device represents for the communication range of IoT device. Therefore, only when OFN enters into this grey circle, IoT device can offload its task to OFN. Meanwhile, this paper assumes continuous time model where the specific events can occur at any time. For example, in Fig. 1, offloading to OFN is possible between $\tau_1$ and $\tau_2$ (i.e., contact time). Thus, if the task of IoT device occurs between $\tau_1$ and $\tau_2$, it immediately offloads the task to OFN without a delay timer. Otherwise, when the task occurs before $\tau_1$, IoT device cannot immediately use OFN. Thus, it delays the offloading up to a pre-defined delay timer $t_D$. Before the delay timer expires, if OFN becomes available owing to its mobility, the pending task can be delivered to OFN. If the task is completed by OFN before $\tau_2$, IoT device can get the result of the task from OFN. However, if the task cannot be completed by OFN during the contact time with IoT device, the task can be offloaded from IoT device to FN again. Then, IoT device can receive the result from it. Note that the disconnection with OFN can be recognized by IoT device by means of the periodic signaling messages [8], [16]. Meanwhile, when OFN is still not available even after the delay timer expires, the task is delivered from IoT device to FN.

For the description of the system model and performance analysis, important notations are described in Table I. We assumed that the contact time between OFN and IoT device, $t_{OFN}$, follows an exponential distribution whose mean value is $1/\mu_{OFN}$ as it is widely exploited for the OFN residence time of the vehicles [16], [17] and mobile users [15]. The probability density function (PDF) and cumulative density function (CDF) can be represented as

$$f_{OFN}(t) = \mu_{OFN} e^{-\mu_{OFN} t} \tag{1}$$

and

$$F_{OFN}(t) = 1 - e^{-\mu_{OFN} t}. \tag{2}$$

In addition, a general distribution with mean $1/\mu_{FN}$ is assumed for the period when OFN is not available, $t_{FN}$ (i.e., non-contact time) [15].

Meanwhile, we assume that the task processing times of FN and OFN follow the exponential distribution as it is widely utilized in the previous fog computing systems [18], [19]. The task processing time of FN, $t_{s,FN}$, follows the exponential distribution with mean $\omega v_{FN}/u_{FN}$ where $\omega$, $v_{FN}$, and $u_{FN}$ are the mean size of the task, computation intensity, and computing capacity of FN, respectively [18]. Then, the PDF and CDF for the task processing time of FN are denoted by

$$f_{s,FN}(t) = \frac{u_{FN}}{\omega v_{FN}} e^{-\frac{u_{FN}}{\omega v_{FN}} t} \tag{3}$$

and

$$F_{s,FN}(t) = 1 - e^{-\frac{u_{FN}}{\omega v_{FN}} t}. \tag{4}$$

Similarly, the exponential distribution with mean $\omega v_{OFN}/u_{OFN}$ is assumed for the task processing time of OFN, $t_{s,OFN}$, where $v_{OFN}$ and $u_{OFN}$ are the computation intensity and computing capacity of OFN [18]. The PDF and CDF of $t_{s,OFN}$ are given by

$$f_{s,OFN}(t) = \frac{u_{OFN}}{\omega v_{OFN}} e^{-\frac{u_{OFN}}{\omega v_{OFN}} t} \tag{5}$$

and

$$F_{s,OFN}(t) = 1 - e^{-\frac{u_{OFN}}{\omega v_{OFN}} t}. \tag{6}$$

Furthermore, the delay timer, $t_D$, is assumed to follow an exponential distribution with mean $1/\lambda_D$ [19]. Then, the PDF and CDF of $t_D$ can be represented as

$$f_D(t) = \lambda_D e^{-\lambda_D t} \tag{7}$$

and

$$F_D(t) = 1 - e^{-\lambda_D t}. \tag{8}$$

By means of the exponential delay timer, the fixed optimal timer can be derived and utilized practically instead of the exponential delay timer [20], [21].

TABLE I
Summary of Notations

| Notation | Description |
|---|---|
| $t_{OFN}$ | OFN contact time |
| $1/\mu_{OFN}$ | Mean value of OFN contact time |
| $t_{FN}$ | FN contact time |
| $1/\mu_{FN}$ | Mean value of FN contact time |
| $t_D$ | Delay timer |
| $1/\lambda_D$ | Mean value of the delay timer |
| $v_{FN}$ | Computation intensity of FN |
| $v_{OFN}$ | Computation intensity of OFN |
| $u_{FN}$ | Computing capacity of FN |
| $u_{OFN}$ | Computing capacity of OFN |
| $\omega$ | Mean size of the task |
| $E$ | Opportunistic offloading probability |
| $P_D$ | Task completion time outage probability |
| $\alpha$ | Target probability |
| $T_C$ | Task completion time |
| $D_t$ | Task completion deadline |

## III. Performance Analysis

In this section, in order to evaluate the performance of DYSON compared to that of the on-the-spot offloading system,[3] the analytic models for both systems are provided. Specifically, the opportunistic offloading probability, which is defined as the probability that the task of IoT device can be offloaded to OFN, is derived as an important performance metric of each system.

[3]In the on-the-spot offloading system, IoT device decides the offloading node (i.e., FN or OFN) according to the current availability of OFN. That is, when the task occurs but OFN is not available, the task is offloaded from IoT device to FN without any delay timer. On the other hand, if OFN is available, IoT device offloads its task to OFN.

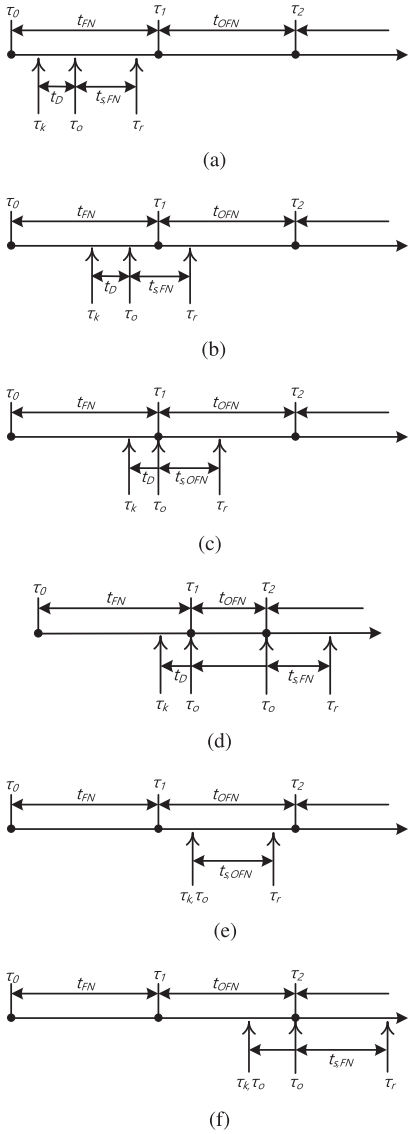


Fig. 2. Timing diagrams of DYSON. (a) $\tau_k, \tau_k + t_D, \tau_r < \tau_1$. (b) $\tau_k, \tau_k + t_D < \tau_1 \leq \tau_r$. (c) $\tau_k < \tau_1 \leq \tau_k + t_D, \tau_r \leq \tau_2$. (d) $\tau_k < \tau_1 \leq \tau_k + t_D \leq \tau_2 < \tau_r$. (e) $\tau_1 \leq \tau_k, \tau_r \leq \tau_2$. (f) $\tau_1 \leq \tau_k \leq \tau_2 < \tau_r$.

### A. Dyson

To obtain the opportunistic offloading probability of DYSON, six disjoint cases need to be considered in Fig. 2: (a) $\tau_k$, $\tau_k + t_D$, $\tau_r < \tau_1$, (b) $\tau_k, \tau_k + t_D < \tau_1 \leq \tau_r$, (c) $\tau_k < \tau_1 \leq \tau_k + t_D, \tau_r \leq \tau_2$, (d) $\tau_k < \tau_1 \leq \tau_k + t_D \leq \tau_2 < \tau_r$, (e) $\tau_1 \leq \tau_k, \tau_r \leq \tau_2$, and (f) $\tau_1 \leq \tau_k \leq \tau_2 < \tau_r$ where $\tau_k, \tau_o$, and $\tau_r$ are the times when the task occurs in IoT device, IoT device offloads its task, and IoT device receives the result of the task, respectively. As mentioned above, $t_{s,FN}$ and $t_{s,OFN}$ denote the task processing times of FN and OFN, respectively. When we let $E_{Y,D}$ be the partial opportunistic offloading probability in the case $Y \in \{\text{a, b, c, d, e, f}\}$ for DYSON, it can be derived as follows.

*1)* $E_{\text{a},D}$*:* When $\tau_k, \tau_k + t_D, \tau_r < \tau_1$, the task is processed by FN because OFN is not available even after the delay timer $t_D$. Therefore, $E_{\text{a},D}$ is simply 0.

*2)* $E_{\text{b},D}$*:* When $\tau_k, \tau_k + t_D < \tau_1 \leq \tau_r$, $E_{\text{b},D}$ is also simply 0 because IoT device offloads its task to FN before OFN is available at $\tau_1$.

$$E_{\text{c},D} = \int_{t_{OFN}=0}^{\infty} \int_{\tau_k=t_{FN}-t_D}^{t_{FN}} \int_{t_{s,OFN}=0}^{t_{OFN}} f_s(t_{s,OFN}) f_k(\tau_k) f_{OFN}(t_{OFN})\, dt_s\, d\tau_k\, dt_{OFN} \tag{9}$$

$$E_{\text{e},D} = \int_{t_{OFN}=0}^{\infty} \int_{\tau_k=t_{FN}}^{t_{FN}+t_{OFN}} \int_{t_{s,OFN}=0}^{t_{FN}+t_{OFN}-\tau_k} f_s(t_{s,OFN}) f_k(\tau_k) f_{OFN}(t_{OFN})\, dt_s\, d\tau_k\, dt_{OFN} \tag{10}$$

*3)* $E_{\text{c},D}$*:* When $\tau_k < \tau_1 \leq \tau_k + t_D, \tau_r \leq \tau_2$, thanks to the delay timer, the offloading to OFN can be possible although the task occurs before OFN is available. In addition, IoT device can get the result of the task from OFN because OFN completes the task within the contact time. We can calculate the partial opportunistic offloading probability in case c by means of the PDF of $t_{s,OFN}$, $\tau_k$, and $t_{OFN}$. The joint PDF of the $t_{s,OFN}$, $\tau_k$, and $t_{OFN}$ is denoted by $f_{s,k,OFN}(t_{s,OFN}, \tau_k, t_{OFN})$. Since $t_{s,OFN}$, $\tau_k$, and $t_{OFN}$ are independent, the joint PDF is the product of each PDF (i.e., $f_s(t_{s,OFN}) f_k(\tau_k) f_{OFN}(t_{OFN})$). As a result, $E_{\text{c},D}$ is given by (9) as shown in the top of the next page.

*4)* $E_{\text{d},D}$*:* When $\tau_k < \tau_1 \leq \tau_k + t_D \leq \tau_2 < \tau_r$, although the offloading to OFN can be possible owing to the delay timer, IoT device cannot get the result of the task because the contact time between OFN and IoT device is not enough for the task processing (i.e., OFN moves away from the communication range of IoT device before it completes the task processing). In this case, IoT device can offload the task to FN again and receive the result from it. Therefore, $E_{\text{d},D}$ is 0.

*5)* $E_{\text{e},D}$*:* Because $\tau_1 \leq \tau_k$ and $\tau_r \leq \tau_2$ represent that the task of IoT device can be delivered to OFN and IoT device can get the result from it, $E_{\text{e},D}$ can be obtained by (10) as shown in the top of the current page.

*6)* $E_{\text{f},D}$*:* The case, $\tau_1 \leq \tau_k \leq \tau_2 < \tau_r$, represents that, even though the task occurs when OFN is available, IoT device cannot get the result of the task because OFN moves away from the communication range of IoT device before it completes the task processing. Thus, as similar with case d, the task is offloaded to FN again and IoT device receives the result from it. Therefore, $E_{\text{f},D}$ is 0.

Since the cases from a to f are disjoint events, the opportunistic offloading probability, $E$, for DYSON can be obtained by $E_{\text{a},D} + E_{\text{b},D} + E_{\text{c},D} + E_{\text{d},D} + E_{\text{e},D} + E_{\text{f},D}$.

On the other hand, even though the longer delay timer means the more opportunities to meet OFN, it leads to the longer task completion time which can directly result in the quality of service (QoS) degradation. Note that QoS of the task offloading service can be generally guaranteed by processing the task within a certain deadline instead of processing the task unconditionally quickly [22]. Therefore, the optimal delay timer is required to maximize the opportunistic offloading probability while maintaining the probability that the task cannot be processed within the deadline (i.e., outage probability, $P_D$) below a target probability $\alpha$.[4] In this paper, the optimal delay timer is derived based on $P_D$ for the worst case (i.e., case d in Fig. 2). Then, this optimal delay timer can maintain $P_D$ below $\alpha$ for all other cases. As $t_D$ is inversely proportional to $\lambda_D$, the optimal rate of $t_D$, $\lambda_D^*$, can be obtained by

$$\begin{aligned} \lambda_D^* &= \underset{\lambda_D}{\text{argmin}}(P_D \leq \alpha) \\ &= \underset{\lambda_D}{\text{argmin}}(P[t_D + t_{OFN} + t_{s,FN} > D_t] \leq \alpha) \end{aligned} \tag{11}$$

[4] $\alpha$ can be determined by the QoS requirements of the task. If the task is delay-sensitive, $\alpha$ should be a small value. Otherwise, $\alpha$ can be a large value.

where argmin() returns the minimum argument satisfying the given inequality. Let $Z$ be the sum of $t_D$, $t_{OFN}$, and $t_{s,FN}$. Since $t_D$, $t_{OFN}$, and $t_{s,FN}$ follow the exponential distribution with mean $1/\lambda_D$, $1/\mu_{OFN}$, and $\omega v_{FN}/u_{FN}$, respectively, PDF and CDF of $Z$ can be represented as [23]

$$f_Z(t) = C\left(\left(\lambda_D - \mu_{OFN}\right)e^{-\frac{u_{FN}}{\omega v_{FN}}t} + \left(\mu_{OFN} - \frac{u_{FN}}{\omega v_{FN}}\right)e^{-\lambda_D t} - \left(\lambda_D - \frac{u_{FN}}{\omega v_{FN}}\right)e^{-\mu_{OFN}t}\right) \tag{12}$$

and

$$F_Z(t) = C\left(\frac{\lambda_D - \frac{u_{FN}}{\omega v_{FN}}}{\mu_{OFN}}e^{-\mu_{OFN}t} - \frac{\mu_{OFN} - \frac{u_{FN}}{\omega v_{FN}}}{\lambda_D}e^{-\lambda_D t} - \frac{\lambda_D - \mu_{OFN}}{\frac{u_{FN}}{\omega v_{FN}}}e^{-\frac{u_{FN}}{\omega v_{FN}}t}\right) \tag{13}$$

where

$$C = \frac{-\lambda_D \mu_{OFN}\frac{u_{FN}}{\omega v_{FN}}}{\left(\lambda_D - \mu_{OFN}\right)\left(\lambda_D - \frac{u_{FN}}{\omega v_{FN}}\right)\left(\mu_{OFN} - \frac{u_{FN}}{\omega v_{FN}}\right)}, \tag{14}$$

respectively. By exploiting (13), (11) can be rearranged as

$$\lambda_D^* = \underset{\lambda_D}{\text{argmin}}\left(1 - F_Z\left(D_t\right) \leq \alpha\right). \tag{15}$$

Based on (15), the optimal timer $\lambda_D^*$ can be derived using an iterative method such as the Newton's method [24]. Note that the delay timer can be 0 when there is no $\lambda_D$ which satisfies (15). This means that $D_t$ and/or $\alpha$ are too tight to determine $\lambda_D$. On the other hand, in the case of the delay tolerable tasks, the delay timer can be infinity.

### B. On-the-Spot Offloading System

Similarly, the opportunistic offloading probability of the on-the-spot offloading system can be obtained by considering four cases in Fig. 3: a) $\tau_k, \tau_r < \tau_1$, b) $\tau_k < \tau_1 \leq \tau_r$, c) $\tau_1 \leq \tau_k, \tau_r \leq \tau_2$, and d) $\tau_1 \leq \tau_k \leq \tau_2 < \tau_r$. Note that, since the task of IoT device is immediately offloaded in both DYSON and the on-the-spot offloading system, the cases c and d of the on-the-spot offloading system are same to those of e and f for DYSON, respectively. Let $E_{X,O}$ be the partial opportunistic offloading probability in the case $X \in \{$ a, b, c, d $\}$ for the on-the-spot offloading system. Then, $E_{X,O}$ can be derived as follows.

*1)* $E_{\mathrm{a},O}$*:* When $\tau_k, \tau_r < \tau_1$, the task is processed by FN because OFN is not available yet. Therefore, $E_{\mathrm{a},O}$ is simply 0.

*2)* $E_{\mathrm{b},O}$*:* When $\tau_k < \tau_1 \leq \tau_r$, $E_{\mathrm{b},O}$ is also simply 0 because IoT device offloads its task to FN.

*3)* $E_{\mathrm{c},O}$*:* As mentioned above, the case c is same to case e in DYSON. Consequently, $E_{\mathrm{c},O}$ can be obtained from (10).

*4)* $E_{\mathrm{d},O}$*:* As mentioned above, the case d is same to case f in DYSON. This means that $E_{\mathrm{d},O}$ is simply 0.

Since the above cases are disjoint, the opportunistic offloading probability, $E$, for the on-the-spot offloading system can be obtained from $E_{\mathrm{a},O} + E_{\mathrm{b},O} + E_{\mathrm{c},O} + E_{\mathrm{d},O}$.

## IV. Simulation Results

To verify the derived analytical models, extensive simulation by means of a developed event-driven simulator was conducted. The default values of $\mu_{OFN}$, $\mu_{FN}$, $\omega$, $u_{OFN}$, $u_{FN}$, $v_{OFN}$, and $v_{FN}$ are 100, 10, 1 $MB$, $5\times10^8$ cycles per second, $10\times10^8$ cycles per second, 50 cycles per $MB$, and 100 cycles per $MB$, respectively [18]. Based on the assumed distributions, 20,000 random numbers of $t_{OFN}$, $t_{FN}$, $t_{s,FN}$, and $t_{s,OFN}$ are generated. Then, with the optimal delay timer $1/\lambda_D^*$ which can be obtained based on the Newton's method [24], they are used to calculate the opportunistic offloading probability. Consequently, the average values are used to show the results. Also, task completion deadline and the target probability (i.e., $D_t$ and $\alpha$) are 0.3 and 0.05 ,[5] respectively.

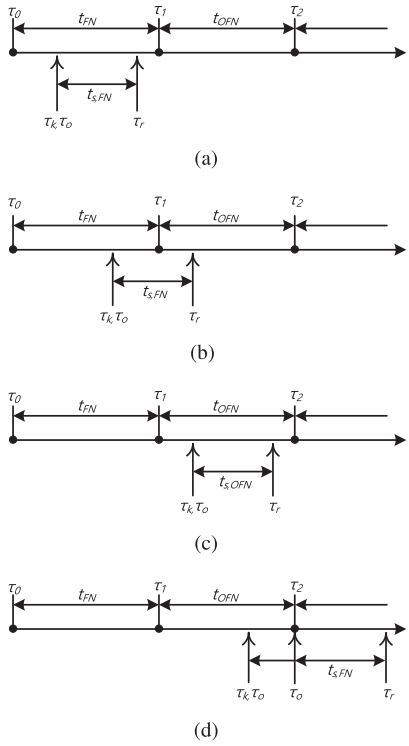


Fig. 3. Timing diagrams of the on-the-spot offloading system. (a) $\tau_k$, $\tau_r < \tau_1$. (b) $\tau_k$, $< \tau_1 \leq \tau_r$. (c) $\tau_1 \leq \tau_k$, $\tau_r \leq \tau_2$. (d) $\tau_1 \leq \tau_k \leq \tau_2 < \tau_r$.

TABLE II
Analytical Results (A) vs. Simulation Results (S)

| $\alpha$ | $1/\lambda_D^*$ | $E(A)$ | $E(S)$ | $T_c(A)$ | $T_c(S)$ |
|---|---|---|---|---|---|
| 0.01 | 0.035 | 0.181 | 0.183 | 0.065 | 0.066 |
| 0.05 | 0.071 | 0.288 | 0.289 | 0.101 | 0.103 |
| 0.1 | 0.101 | 0.400 | 0.406 | 0.131 | 0.133 |

The comparison between the analytical and simulation results is shown in Table II. As shown in Table II, the simulation results are very similar with the analytic results (i.e., less than 2% difference). In addition, from Table II, it can be found that the opportunistic offloading probability (i.e., $E$) and the task completion time (i.e., $T_c$) become higher with the increase of the target probability. This can be explained as follows. When $\alpha$ has a large value (i.e., loose outage probability constraint), the optimal delay timer can be set to larger value as shown in the second column of Table II.

Fig. 4 shows the opportunistic offloading probability $E$ with the contact time $1/\mu_{OFN}$ which is normalized by the non-contact time

[5]Even though there is a probability of the case d (e.g., 5.09% in our default simulation setting), when the optimal delay timer is exploited, most of tasks (e.g., 95% tasks when $\alpha = 0.05$) can be processed within the deadline.

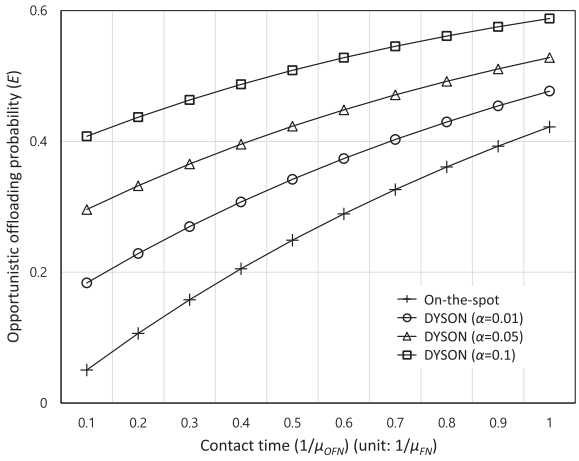


Fig. 4. Opportunistic offloading probability $E$ with the contact time $1/\mu_{OFN}$.

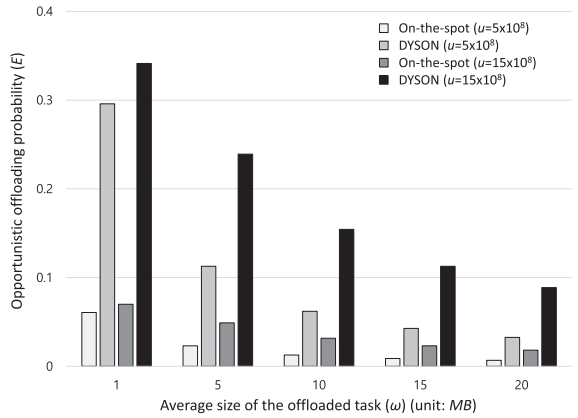


Fig. 5. Opportunistic offloading probability $E$ with the mean size of the task $\omega$.

$1/\mu_{FN}$. Meanwhile, as $1/\mu_{OFN}$ increases, $E$ increases for both systems. This is because as $1/\mu_{OFN}$ increases, which can also mean the decrease of the moving speed of OFN, there is more chance to utilize OFN. Fig. 4 also shows that $E$ of DYSON with the optimal delay timer for any $\alpha$ is higher than that of the on-the-spot offloading system under every $1/\mu_{OFN}$. The on-the-spot offloading system starts offloading immediately when the task occurs while DYSON defers the offloading at most $t_D$. Therefore, the on-the-spot offloading system relatively has little chance to use OFN than that of DYSON, which results in lower $E$. In addition, the efficiency of DYSON becomes apparent as $\alpha$ increases. For example, with the setting $1/\mu_{OFN} = 0.6$ (i.e., the sixth x-axis point), $E$ of DYSON is improved by 121%, 149%, and 175% compared to the on-the-spot offloading system, when $\alpha = 0.01$, 0.05, and 0.1, respectively. On the other hand, it can be found that the efficiency of DYSON decreases as $1/\mu_{OFN}$ increases. For example, the improvement rate of $E$ of DYSON compared to the on-the-spot offloading system is 298% when $1/\mu_{OFN}$ is 0.1. However, it becomes 103% when $1/\mu_{OFN}$ is 1. The reason is that the optimal delay timer becomes reduced according to the increasing contact time, which decreases the effect of the delay.

Fig. 5 shows the opportunistic offloading probability $E$ with the mean size of the task $\omega$ when $\alpha = 0.05$. It is found that $E$ decreases according to the increase of $\omega$ for both systems. The reason is that according to the increase of $\omega$, the task processing time also increases, which reduces the probability that the task can be completed by OFN within the contact time. $E$ is reduced by 62% according to the change of $\omega$ from 1 to 5 when $u = 5\text{x } 10^8$ for both systems. Even though $E$ decreases according to $\omega$ for both systems, DYSON has higher $E$ than that of the on-the-spot offloading system with the same $u$. This is because DYSON has more chance to offload the task to OFN owing to $t_D$ than the on-the-spot offloading system. Specifically, with the setting $\omega = 10$, $E$ of DYSON is improved by 388% and 387% when $u = 5$ x$10^8$ and 15x$10^8$, respectively. In addition, it can be checked from Fig. 5 that $E$ increases when $u$ increases. The reason is that the task processing time can be reduced when $u$ increases.

## V. Conclusion

This paper introduced a DYSON where IoT device can delay the task offloading to maximize the offloading effect of opportunistic fog node (OFN). In addition, the analytic model of the opportunistic offloading probability was developed to assess the performance of the proposed system. Based on the analytic model, we derive the optimal delay timer to maximize the opportunistic offloading probability while maintaining the probability that the task cannot be processed within the deadline below a target probability. Analytic and simulation results demonstrated that DYSON with the optimal delay timer can significantly improve the opportunistic offloading probability compared to the on-the-spot offloading system. In addition, it is shown that the opportunistic offloading probability is influenced by the task processing time, delay timer, task size, and computing capacity. In our future work, this paper will be extended to handle the competition to get the constrained computing resource of OFN considering multiple IoT devices.

## References

[1] H. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 68–76, Mar./Apr. 2015.
[2] Y. Liu *et al.*, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25445–25454, 2017.
[3] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *IEEE Comput.*, vol. 49, no. 8, pp. 112–116, Aug. 2016.
[4] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 253–262, Jan.–Mar. 2020.
[5] X. Wang, Z. Ning, and L. Wang, "Offloading in internet of vehicles: A fog-enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.
[6] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.
[7] N. Fernando *et al.*, "Opportunistic fog for IoT: Challenges and opportunities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8897–8910, Oct. 2019.
[8] C. Zhu *et al.*, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4150–4161, Jun. 2019.
[9] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, "Mobile-edge computing for vehicular networks," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[10] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sep. 2019.
[11] A. Lakhan, M. Ahmad, M. Bilal, A. Jolfaei, and R. M. Mehmood, "Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing," *IEEE Trans. Intell. Transport Syst.*, vol. 22, no. 7, pp. 4212–4223, Jul. 2021.
[12] H. Liao *et al.*, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, Jul. 2021.
[13] H. Ko, J. Lee, and S. Pack, "Performance optimization of delayed WiFi offloading in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9436–9447, Oct. 2017.

[14] H. Wu and K. Wolte, "Stochastic analysis of delayed mobile offloading in heterogeneous networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 461–474, Feb. 2018.
[15] D. Suh, H. Ko, and S. Pac, "Efficiency analysis of WiFi offloading techniques," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3813–3817, May 2016.
[16] Y. Liu, W. Wang, Y. Ma, Z. Yang, and F. Yu, "Distributed task offloading in heterogeneous vehicular crowd sensing," *Sensors*, vol. 16, no. 7, Jul. 2016, Art no. 1090.
[17] J. Lee *et al.*, "Pseudonyms in IPv6 ITS communications: Use of pseudonyms, performance degradation, and optimal pseudonym change," *Int. J. Distrib. Sensor Netw.*, vol. 15, pp. 1–7, May 2015.
[18] J. Yao and N. Ansari, "QoS-Aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 16, no. 1, pp. 167–175, Mar. 2019.
[19] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.
[20] H. Fu, P. Lin, and Y. Lin, "Reducing signaling overhead for femtocell/macrocell networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 8, pp. 1587–1597, Aug. 2013.
[21] T. Katayama, "Analysis of a time-limited service priority queueing system with exponential timer and server vacations," *Queuing Syst.*, vol. 57, no. 4, pp. 169–178, Dec. 2007.
[22] C. Swain *et al.*, "METO: Matching-theory-based efficient task offloading in IoT-Fog interconnection networks," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12705–12715, Aug. 2021.
[23] M. Akkouchi, "On the convolution of exponential distributions," *J. Chungcheong Math. Soc.*, vol. 21, no. 4, pp. 501–510, Dec. 2008.
[24] P. Deuflhard, *Newton Methods for Nonlinear Problems, Affine Invariance and Adaptive Algorithms*. Berlin, Germany: Springer-Verlag, 1994.