

# Performance Analysis of Task Offloading With Opportunistic Fog Nodes

YEUNWOONG KYUNG <sup>1</sup>

School of Computer Engineering, Hanshin University, Osan 18101, South Korea

e-mail: ywkyung@hs.ac.kr

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2020R1G1A1100493.

**ABSTRACT** In the fog computing architecture, the offloading of computing tasks can be conducted by the Internet of Things (IoT) devices to the fog nodes (FNs) that are co-located with base stations (BSs). However, as the IoT devices within the same coverage of the BS can offload lots of tasks simultaneously, the FN can be overloaded, resulting in scalability issues due to limited computing resources. As a promising solution to this problem, opportunistic FNs (OFNs) which denote FNs with mobility such as smart phones and vehicles have been considered as they opportunistically reduce the load of static FNs. IoT devices can offload a task and receive the result to/from the OFN directly when OFN is close to the device. In addition, the offloading can be conducted indirectly through the BS when the OFN is not in the vicinity of the IoT devices while it is within the coverage of the BS. To assess the offloading performance according to the mobility of the OFN considering the direct and indirect offloading scenarios, we developed an analytic model for the opportunistic offloading probability that the task can be offloaded to the OFN, which can also be interpreted as the load distribution effect. Extensive simulation results are given to validate the analytic model and to demonstrate the performance of the opportunistic offloading probability.

**INDEX TERMS** Fog computing, opportunistic fog node, load distribution.

## I. INTRODUCTION

With the growing popularity of smart devices, such as smart phones, wearable devices, and vehicles that are equipped with various sensors, Internet of Things (IoT) networks where data is shared between the connected devices and IoT services such as smart home, smart city, and smart factory have attracted increased attention [1], [2]. Along with the variety and complexity of IoT services, cloud computing has been utilized to process the task from the IoT devices due to its flexible and scalable computing resources [3]. However, the remote location of the cloud servers typically leads to high latency and requires high network bandwidth usage for the core network. To mitigate this problem, the concept of fog computing, which brings computing resources closer to the IoT devices, has been introduced [4], [5].

In the fog computing architecture, the offloading of computing tasks can be conducted from the IoT devices to the fog node (FN) that is usually co-located with the network

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Casoni <sup>1</sup>.

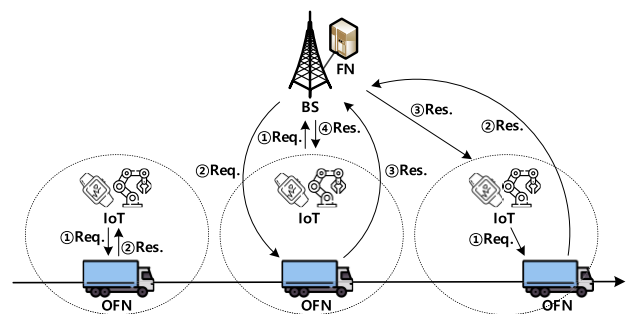


FIGURE 1. Offloading scenarios: (a) direct, (b) indirect, (c) hybrid.

access node such as a base station (BS) [6]. Compared to the conventional offloading (i.e., based on the cloud computing), offloading with FN can reduce the load of networks and support delay-sensitive applications. However, as the IoT devices within the same coverage of the BS can offload lots of tasks simultaneously, the FN can be overloaded, resulting in scalability issues due to its limited computing resource [6]–[9].

As a promising solution to this problem, opportunistic FNs (OFNs) which denote the FNs with mobility such as smart phones and vehicles have been considered. OFNs can provide computing resources to the IoT devices only when they are close to each other. In other words, OFNs can opportunistically reduce the load of static FNs because of mobility [10]–[19]. There have been many studies on the OFN offloading that can be classified into three scenarios as shown in Fig. 1: (a) direct offloading where the IoT devices directly offload the task and receive the result to/from the OFN when the OFN is close to the IoT devices [10]–[12], (b) indirect offloading where both procedures (i.e., the task offloading and result reception) always pass through the BS when the OFN is not in the vicinity of IoT devices while it is within the coverage of BS [13]–[16], and (c) hybrid offloading where either of the procedures directly interacts with OFN while the other is conducted through the BS according to the availability of the OFN based on its mobility [17]–[19]. However, previous studies have only considered their own fixed offloading scenarios. This means that the overall offloading analysis that reflects the entirety of the scenarios has not been considered even though all the scenarios can be opportunistically utilized according to the mobility of the OFN.

In this paper, we focus on the challenges associated with analytically modeling the behavior of the OFN offloading based on the whole systematic view. Based on the analytical model, extensive simulations are conducted and demonstrate that direct and hybrid offloadings account for 73% of the total opportunistic offloading probability, which denotes that the direct and hybrid offloading scenarios should be considered to design the OFN-based offloading architecture. To the best of our knowledge, this is the first analytical study that elaborates on the performance of the OFN offloading considering the entirety of the available scenarios according to the mobility of the OFN under diverse environments. The key contribution of this paper is two-fold: 1) we develop an analytic model for the opportunistic offloading probability that a task can be offloaded to the OFN for each offloading scenario, which can be interpreted as the load distribution effect; and 2) based on the simulation works, we evaluate the performance of the opportunistic offloading probability under various environments, which can provide valuable design guidelines for the OFN-based offloading architectures.

The remainder of this paper is organized as follows. After related works are introduced in Section II, the system model is described in Section III. Performance analysis is conducted in Section IV. Evaluation results and concluding remarks are given in Sections V and VI, respectively.

## II. RELATED WORKS

Due to the limited computing resource of FN, there have been efforts to optimize the offloading performance between FN and users [6]–[9]. Fan *et al.* [6] proposed a workload balancing scheme to minimize the service latency considering traffic load allocation to BS and computing load allocation to FN. In addition, an application-aware load distribution

scheme was introduced that can dynamically adjust computing resources of different applications to FN [7]. Ali *et al.* [8] provided an energy-efficient deep learning based computation offloading scheme to minimize the cost of local and remote executions considering the remaining energy of users, previous offloading decision, and network conditions. Moreover, Irshard *et al.* [9] considered a wireless powered FN system where users can offload the task to FN and harvest energy from FN, and provided an optimal time allocation and offloading policies to minimize the cost and energy consumption of users. These efforts on the offloading with FN have been extended to consider OFN.

As introduced in Section I, lots of studies have been conducted to opportunistically reduce the load of static FNs using OFNs [10]–[19]. These studies can be categorized into the direct offloading between the IoT devices and the OFN [10]–[12], indirect offloading via the BS [13]–[16], and hybrid offloading that considers both direct and indirect offloadings [17]–[19]. Shah *et al.* [10] introduced a vehicular fog computing framework to consider computational offloading from user vehicles to other fog vehicles and provide a scheduling policy for the offloading. Waqas *et al.* [11] investigated the effects of the mobility factors that captures the level of physical proximity between the users and fog nodes to the offloading performances. Lee *et al.* [12] provided a vehicular fog computing system to reduce the service delay of vehicular applications by offloading computational tasks to vehicles based on the resource monitoring manager.

Wang *et al.* [13] considered a fog computing architecture where roadside units (RSUs) redirect the incoming tasks to available vehicles to minimize the average response time. Hou *et al.* [14] introduced a software-defined vehicular fog computing architecture where RSU-based task allocation to vehicles is performed based on the offloading decision by software-defined networking (SDN) controller, while also maintaining the global network state information. Ye *et al.* [15] provided a service offloading strategy in bus networks to find an optimal task allocation strategy according to the moving trajectory of the bus fog servers. Tang *et al.* [16] proposed a task scheduling scheme where RSUs determine the task offloading strategy among vehicular fogs to reduce the service latency.

Zhang *et al.* [17] introduced a hybrid offloading scheme where either the vehicular-to-infrastructure (V2I) or vehicular-to-vehicular (V2V) transmissions is determined as task offloading options to reduce the offloading cost while considering running direction, file transmission time, and task execution time. Zhou *et al.* [18] reviewed the hybrid usage of the V2V and V2I offloadings in the vehicular fog computing architecture from the perspective of the mobility feature such as the contact probability, contact time interval, and coded computing. Wu *et al.* [19] provided a fog node selection algorithm to minimize the average response time of the offloaded tasks utilizing both V2V and V2I communications.

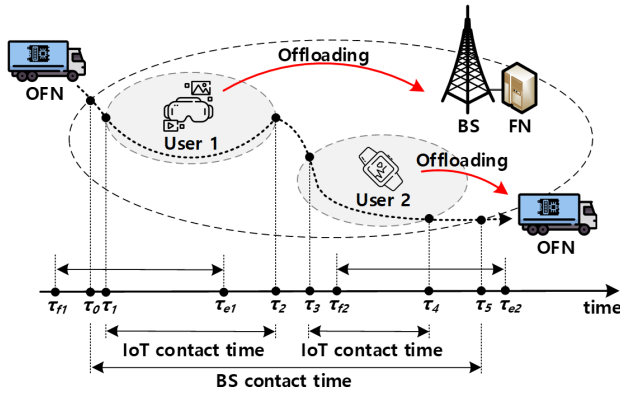


FIGURE 2. System model.

Note that abovementioned previous studies focused on their own offloading scenarios, which means that it was not investigated to analyze the OFN offloading performance considering the whole available offloading scenarios.

### III. SYSTEM MODEL

As an example, in Fig. 2, the OFN can be in contact with the BS between  $\tau_0$  and  $\tau_5$  (i.e., the BS contact time). In the BS contact time, the OFN can communicate with IoT devices (i.e., users 1 and 2) between  $\tau_1$  and  $\tau_2$  and between  $\tau_3$  and  $\tau_4$  (i.e., the IoT contact times), respectively. If the task of the users occurs during the IoT contact time (e.g.,  $\tau_{f2}$ ), the IoT device can directly offload the task to the OFN. Then, the IoT device can receive the result of the task from the OFN if the OFN completes the task before  $\tau_4$ . This case does not impose any load on the BS. However, it can be possible that the OFN cannot complete the task (e.g.,  $\tau_{e2}$ ) during the IoT contact time (e.g., between  $\tau_3$  and  $\tau_4$ ). In this case, the OFN can deliver the task result via the BS if the task is completed before  $\tau_5$ . On the other hand, if the task occurs before or after the IoT contact time (e.g.,  $\tau_{f1}$ ), the IoT device transmits the task to BS because it cannot directly use the OFN. From the BS's perspective, if the task is received during the BS contact time, the BS can offload the task to the OFN because the OFN is available. If the OFN cannot complete the task before  $\tau_5$ , this paper assumes that the result can be delivered to the original BS based on the interface between the BSs such as X2 and Xn interfaces [20], [21]. Similarly, if the task occurs at the IoT device before  $\tau_0$  or after  $\tau_5$ , the IoT device offloads the task to the FN through the BS. Note that the IoT device and the BS can recognize the disconnection with the OFN by using periodic signaling messages [22], [23].

It is assumed that the IoT and BS contact times denoted by  $t_I$  and  $t_B$  follow an exponential distribution with mean  $1/\mu_I$  and  $1/\mu_B$  [24], [25], respectively. The probability density function (PDF) and cumulative density function (CDF) for the IoT contact time are denoted by

$$f_I(t) = \mu_I e^{-\mu_I t} \quad (1)$$

and

$$F_I(t) = 1 - e^{-\mu_I t}. \quad (2)$$

Similarly, the probability density function (PDF) and cumulative density function (CDF) for BS contact time are denoted by

$$f_B(t) = \mu_B e^{-\mu_B t} \quad (3)$$

and

$$F_B(t) = 1 - e^{-\mu_B t}. \quad (4)$$

In addition, we assume that the period when the OFN is not available for both the IoT and BS (i.e., non-contact time), denoted by  $t_N$ , follows a general distribution with mean  $1/\mu_N$  [27].

Meanwhile, it is assumed that the task processing time of the OFN denoted by  $t_p$  follows the exponential distribution with the mean  $\omega v_{OFN}/u_{OFN}$ , where  $\omega$ ,  $v_{OFN}$ , and  $u_{OFN}$  are the average size of the offloaded task, computational intensity, and computing capacity of OFN, respectively [28]. Its PDF and CDF are given by

$$f_p(t) = \frac{u_{OFN}}{\omega v_{OFN}} e^{-\frac{u_{OFN}}{\omega v_{OFN}} t} \quad (5)$$

and

$$F_p(t) = 1 - e^{-\frac{u_{OFN}}{\omega v_{OFN}} t} \quad (6)$$

### IV. PERFORMANCE ANALYSIS

In this section, analytic models are introduced to evaluate the performance of the opportunistic offloading probability that task is offloaded to the OFN. Specifically, available OFN offloading scenarios can be divided into six disjoint cases in Fig. 3: a)  $\tau_1 \leq \tau_k, \tau_r < \tau_2$ , b)  $\tau_1 \leq \tau_k < \tau_2 \leq \tau_r \leq \tau_3$ , c)  $\tau_1 \leq \tau_k < \tau_2, \tau_3 < \tau_r \leq \tau_4$ , d)  $\tau_2 \leq \tau_k, \tau_r \leq \tau_3$ , e)  $\tau_2 \leq \tau_k \leq \tau_3 < \tau_r \leq \tau_4$ , and f)  $\tau_3 < \tau_k, \tau_r \leq \tau_4$ , where  $\tau_k$  and  $\tau_r$  are the times when the task occurs in the IoT device and the result of the task is received, respectively.  $t_p$  denotes task processing time of the OFN. As described in Figure 3, we assume that the BS and IoT contact times can have range from 0 to infinity (i.e.,  $0 \leq t_B \leq \infty$ ) and from 0 to the BS contact time (i.e.,  $0 \leq t_I \leq t_B$ ), respectively. In addition, the IoT device can meet the OFN when they are both in the coverage of BS (i.e.,  $t_N \leq \tau_2 \leq t_N + t_B - t_I$ ). As explained before, we do not consider the cases when  $\tau_k < \tau_1, \tau_4 < \tau_k$ , and  $\tau_4 < \tau_r$ , because the FN (i.e., not OFN) is in charge of the offloaded task in these cases.

Let  $E_Y$  be the opportunistic offloading probability in the case  $Y \in \{a, b, c, d, e, f\}$ . Then,  $E_Y$  can be derived as follows.

- 1)  $E_a$ : As shown in Figure 3(a), when  $\tau_1 \leq \tau_k, \tau_r < \tau_2$ , the task is delivered to the OFN through the BS because the OFN is not within the IoT contact time (i.e.,  $t_I$ ) but within the BS contact time (i.e.,  $t_B$ ). This means that the task occurs before the IoT device meets the OFN during the BS contact time (i.e.,  $t_N \leq \tau_k \leq \tau_2$ ). In addition, the result of the task can be received from

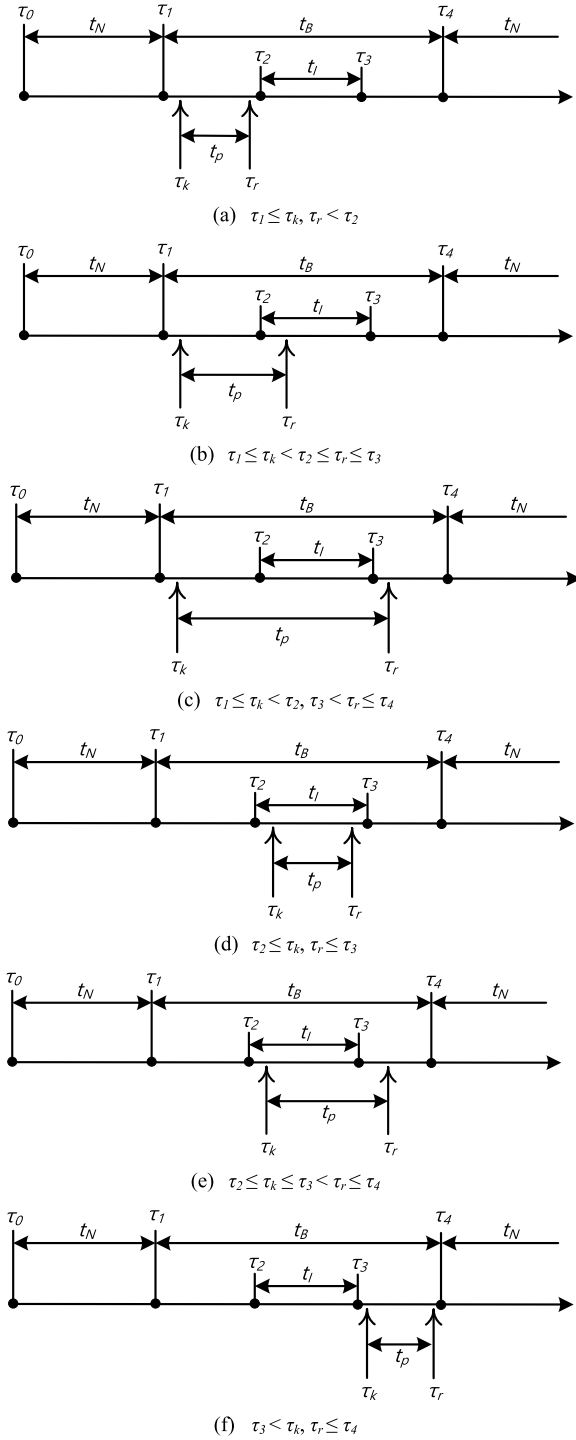


FIGURE 3. Timing diagrams of OFN offloading cases.

the BS because the task is completed before the IoT contact time (i.e.,  $0 \leq t_p \leq \tau_2 - \tau_k$ ). The opportunistic offloading probability in case a (i.e.,  $E_a$ ) can be calculated using the PDF of  $t_p$  and  $\tau_k$ . The joint PDF of the  $t_p$  and  $\tau_k$  is denoted by  $f_{p,k}(t_p, \tau_k)$ . Since  $t_p$  and  $\tau_k$  are independent, the joint PDF becomes the product of each PDF (i.e.,  $f_p(t_p) \cdot f_k(\tau_k)$ ). Consequently,

$E_a$  is given by

$$E_a = \int_{\tau_k=t_N}^{\tau_2} \int_{t_p=0}^{\tau_2-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (7)$$

- 2)  $E_b$ : When  $\tau_1 \leq \tau_k < \tau_2 \leq \tau_r \leq \tau_3$ , after the IoT device offloads the task to the OFN through the BS (i.e.,  $t_N \leq \tau_k \leq \tau_2$ ), it can receive the result of the task directly from the OFN because the task is completed within the IoT contact time (i.e.,  $\tau_2 - \tau_k \leq t_p \leq \tau_2 + t_I - \tau_k$ ). This means that OFN moves close to the IoT device before the task is completed. Consequently,  $E_b$  is given by

$$E_b = \int_{\tau_k=t_N}^{\tau_2} \int_{t_p=\tau_2-\tau_k}^{\tau_2+t_I-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (8)$$

- 3)  $E_c$ : When  $\tau_1 \leq \tau_k < \tau_2, \tau_3 < \tau_r \leq \tau_4$ , after the IoT device offloads the task to the OFN through the BS (i.e.,  $t_N \leq \tau_k \leq \tau_2$ ), it can also receive the result of the task from the OFN through the BS because the task is completed after the IoT contact time (i.e.,  $\tau_2 + t_I - \tau_k \leq t_p \leq t_N + t_B - \tau_k$ ). This means that OFN passes by the IoT device while processing the task. Consequently,  $E_c$  is given

$$E_c = \int_{\tau_k=t_N}^{\tau_2} \int_{t_p=\tau_2+t_I-\tau_k}^{t_N+t_B-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (9)$$

- 4)  $E_d$ : When  $\tau_2 \leq \tau_k, \tau_r \leq \tau_3$ , the IoT device directly offloads to OFN because the task occurs during the IoT contact time (i.e.,  $\tau_2 \leq \tau_k \leq \tau_2 + t_I$ ). In addition, the IoT device receives the result of the task to/from the OFN because the OFN is within the IoT contact time from the task occurrence time to the end of the processing time (i.e.,  $0 \leq t_p \leq \tau_2 + t_I - \tau_k$ ). Consequently,  $E_d$  is given by

$$E_d = \int_{\tau_k=\tau_2}^{\tau_2+t_I} \int_{t_p=0}^{\tau_2+t_I-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (10)$$

- 5)  $E_e$ : When  $\tau_2 \leq \tau_k \leq \tau_3 < \tau_r \leq \tau_4$ , although the offloading to the OFN can be directly possible from the IoT device (i.e.,  $\tau_2 \leq \tau_k \leq \tau_2 + t_I$ ), the result of the task is delivered to the IoT device from the OFN via the BS because the IoT contact time is not sufficient to complete the task (i.e., the OFN leaves from the coverage of the IoT device before the task processing is completed;  $\tau_2 + t_I - \tau_k \leq t_p \leq t_N + t_B - \tau_k$ ). Therefore,  $E_e$  is given by

$$E_e = \int_{\tau_k=\tau_2}^{\tau_2+t_I} \int_{t_p=\tau_2+t_I-\tau_k}^{t_N+t_B-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (11)$$

- 6)  $E_f$ : Similar to the case a, when  $\tau_3 < \tau_k, \tau_r \leq \tau_4$ , the task is offloaded, and the result of the task is received after the IoT contact time (i.e.,  $\tau_2 + t_I \leq \tau_k \leq t_N + t_B$  and  $0 \leq t_p \leq t_N + t_B - \tau_k$ ). Consequently,  $E_f$  is given by

$$E_f = \int_{\tau_k=\tau_2+t_I}^{t_N+t_B} \int_{t_p=0}^{t_N+t_B-\tau_k} f_p(t_p) f_k(\tau_k) dt_p d\tau_k. \quad (12)$$

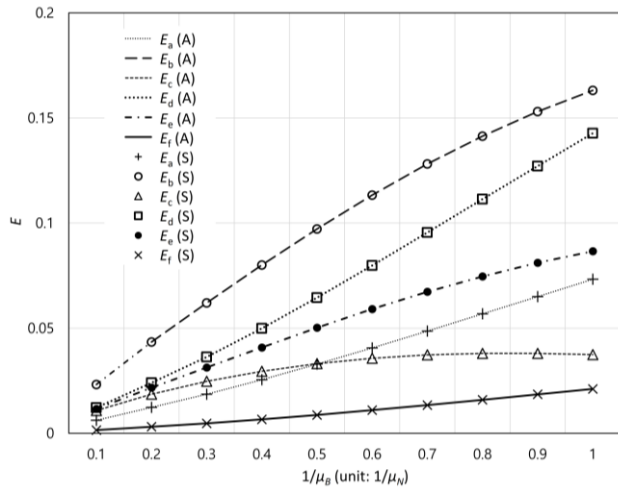


FIGURE 4. Opportunistic offloading probability  $E$  for each case according to the BS contact time  $1/\mu_B$ .

Since the cases from a to f are disjoint events, the total opportunistic offloading probability denoted by  $E_O$  can be obtained from (13).

$$E_O = E_a + E_b + E_c + E_d + E_e + E_f \quad (13)$$

### V. SIMULATION RESULTS

To validate the analytic models, we developed an event-driven simulator based on MATLAB R2018a and conducted extensive simulations. The default values of  $\mu_N$ ,  $\mu_B$ , and  $\mu_I$  are 5, 5, and  $\mu_B \times 2$ , respectively, as the average contact time of OFN is usually assumed to have a range from 1 to 10 [24]–[26]. In addition, the default values of  $\omega$ ,  $\omega_{OFN}$ , and  $\nu_{OFN}$  are set to 1Mb,  $5 \times 10^8$  cycles per second, and 50 cycles per Mb, respectively [27]. For the simulation results, 50,000 random numbers are generated for  $t_p$ ,  $t_B$ ,  $t_I$ ,  $\tau_k$ , and  $t_{OFN}$  according to the distribution with the default mean values as described above and used to compute the opportunistic offloading probability. This means that each simulation result is the average value of 50,000 tests.

Fig. 4 shows the opportunistic offloading probability  $E$  for each case according to the BS contact time  $1/\mu_B$  which is normalized by the non-contact time  $1/\mu_N$ . The reason of this normalization is that Fig. 4 aims to show the effect of the BS contact time relative to the non-contact time (i.e., the ratio of the BS contact time to the non-contact time is the x-axis in Fig. 4). First, as shown in Fig. 4, the simulation results are consistent with the analytic results in all simulation settings. Meanwhile, as  $1/\mu_B$  increases,  $E$  increases for all cases. This is because as  $1/\mu_B$  increases, which can also be interpreted that the moving speed of the OFN decreases within the coverage of the BS, there are increased chances to exploit the OFN. Fig. 4 also describes that  $E_b$ ,  $E_d$ , and  $E_e$  are higher than  $E_a$ ,  $E_c$ , and  $E_f$ . Since we assume that the BS contact time is twice of the IoT contact time, the probability that either  $\tau_k$  or  $\tau_r$  exists within the IoT contact time is higher than the probability that

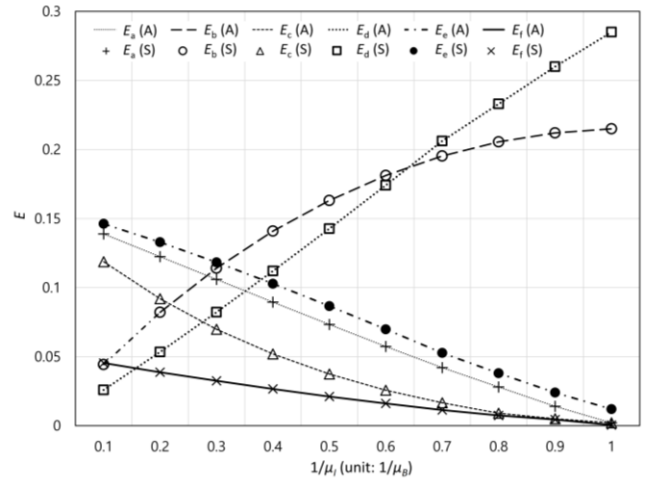
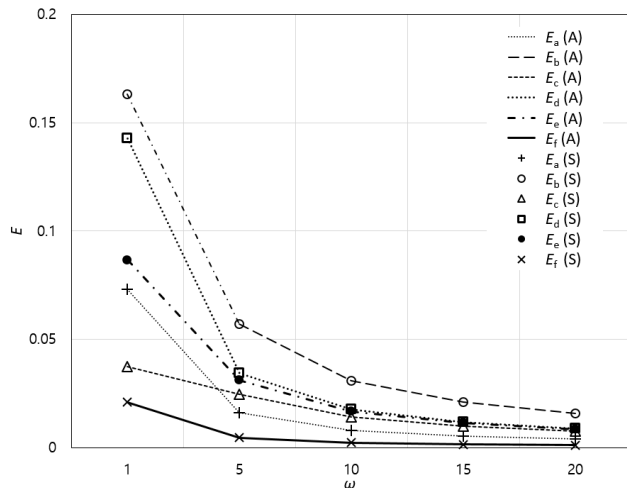


FIGURE 5. Opportunistic offloading probability  $E$  according to the IoT contact time  $1/\mu_I$ .

both do not exist within the IoT contact time. Therefore,  $E_b$ ,  $E_d$ , and  $E_e$ , when either  $\tau_k$  or  $\tau_r$  exists within the IoT contact time, is higher than others. In addition, a crossing point can be found as  $1/\mu_B$  increases. Specifically,  $E_a$  sharply increases and becomes higher than  $E_c$ . This is because the probability of a short processing time is higher than the probability of a long processing time for the OFN based on the form of the distribution. From Fig. 4, we can also check that the direct and hybrid offloadings between the OFN and IoT device can be efficiently utilized. Specifically, when  $1/\mu_B = 0.5$  (i.e., the fifth x-axis point), the direct and hybrid offloadings (i.e., cases b, d, and e) account for 73% of  $E_O$ . This means that the direct and hybrid offloadings should be considered to design the OFN-based offloading architecture.

Fig. 5 shows the opportunistic offloading probability  $E$  for each case according to the IoT contact time  $1/\mu_I$  which is normalized by the BS contact time  $1/\mu_B$ . The reason of this normalization is that Fig. 5 aims to describe the effect of the IoT contact time relative to the BS contact time (i.e., the ratio of the IoT contact time to the BS contact time is the x-axis in Fig. 5). First, as shown in Fig. 5,  $E_b$  and  $E_d$  increase while others decrease as  $1/\mu_I$  increases. This is because the opportunity to complete the task processing within the IoT contact time becomes higher as  $1/\mu_I$  increases. It can be noted that as  $1/\mu_I$  increases, which can also be interpreted that the coverage of dedicated IoT communication increases, there can be more chances to exploit the OFN directly from the IoT. Between the cases b and d,  $E_d$  sharply increases and finally becomes higher than  $E_b$  because it becomes easier to offload the task and receive the result of the task directly to/from the OFN with increasing IoT contact time. On the other hand, among the cases a, c, e, and f,  $E_e$  has the highest value because the task can easily occur within the IoT contact time as  $1/\mu_I$  increases. With other cases (i.e., cases a, c, and f),  $E_a$  is higher than others due to the form of distribution for the task occurrence and processing time as explained above. From



**FIGURE 6.** Opportunistic offloading probability  $E$  according to the average size of the offloaded task  $\omega$ .

Fig. 5, we can also check that the direct offloading between the OFN and IoT device without the intervention of the BS can be efficiently utilized. Specifically, when  $1/\mu_I = 0.7$  (i.e., the seventh x-axis point), 40% of  $E_O$  does not require the intervention of the BS, which can reduce the load of the BS as well as the FN.

Fig. 6 shows the opportunistic offloading probability  $E$  for each case according to the average size of the offloaded task  $\omega$  whose unit is Mb. It is shown that  $E$  decreases as  $\omega$  increases for all cases. This is because the task processing time also increases as  $\omega$  increases, which makes it difficult for the OFN to complete the task within the BS and IoT contact times. Among all cases, the cases utilizing the IoT contact time (i.e., cases b, d, and e) have a higher  $E$  as explained with Fig. 5. On the other hand, among the other cases (i.e., cases a, c, and f), case f has the lowest  $E$  due to the form of the distribution for the task occurrence as explained above. Then, we can see that  $E_a$  becomes lower than  $E_c$  as  $\omega$  increases. This is because it is difficult for the OFN to complete the task within short time (e.g., before the IoT contact time) as  $\omega$  increases.

## VI. CONCLUSION

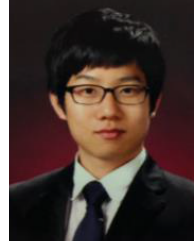
In this paper, we introduced a systematic view to design the OFN-based offloading architecture which can be dynamically flexible according to the mobility of the OFN. In addition, we developed the analytic model of the opportunistic offloading probability considering both direct, indirect and hybrid OFN offloading scenarios based on the mobility of the OFN. Analytic and simulation results demonstrated that the opportunistic offloading probability for each case is affected by the contact times of the OFN with the BS and IoT device as well as the size of the offloaded task. Based on the results, it was found that the opportunistic offloading probability in each case increases when the BS contact time increases and the average size of the offloaded task decreases.

On the other hand, when the IoT contact time increases, the opportunistic offloading probability in the direct and specific hybrid offloading cases increases and accounts for the major parts while the opportunistic offloading probability in the other cases decreases. This denotes that the direct and hybrid offloading scenarios should be considered to design the OFN-based offloading architecture. In our future work, we will investigate the optimal offloading performance considering the available resources of the BS, static FN, and OFN.

## REFERENCES

- [1] S. Roy, D. Puthal, S. Sharma, S. P. Mohanty, and A. Y. Zomaya, "Building a sustainable Internet of Things: Energy-efficient routing using low-power sensors will meet the need," *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, pp. 42–49, Mar. 2018.
- [2] L. Fang, H. Zhang, M. Li, C. Ge, L. Liu, and Z. Liu, "A secure and fine-grained scheme for data security in industrial IoT platforms for smart city," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7982–7990, Sep. 2020.
- [3] H.-L. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 68–76, Mar./Apr. 2015.
- [4] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25445–25454, 2017.
- [5] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.
- [6] 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. 2018.
- [7] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.
- [8] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z. H. Abbas, and S. Khaf, "A deep learning approach for energy efficient computational offloading in mobile edge computing," *IEEE Access*, vol. 7, pp. 149623–149633, 2019.
- [9] A. Irshad, Z. H. Abbas, Z. Ali, G. Abbas, T. Baker, and D. Al-Jumeily, "Wireless powered mobile edge computing systems: Simultaneous time allocation and offloading policies," *Electronics*, vol. 10, no. 8, p. 965, Apr. 2021.
- [10] S. S. Shah, M. Ali, A. W. Malik, M. A. Khan, and S. D. Ravana, "VFog: A vehicle-assisted computing framework for delay-sensitive applications in smart cities," *IEEE Access*, vol. 7, pp. 34900–34909, 2019.
- [11] M. Waqas, Y. Niu, M. Ahmed, Y. Li, D. Jin, and Z. Han, "Mobility-aware fog computing in dynamic environments: Understandings and implementation," *IEEE Access*, vol. 7, pp. 38867–38879, 2018.
- [12] S.-S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020.
- [13] 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.
- [14] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang, "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7097–7111, Aug. 2020.
- [15] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CS3Cloud)*, Jun. 2016, pp. 247–251.
- [16] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9364–9375, Sep. 2020.
- [17] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

- [18] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 49–55, May 2019.
- [19] Y. Wu, J. Wu, L. Chen, G. Zhou, and J. Yan, "Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2599–2614, May 2021.
- [20] *LTE; Overall Description (Release 16)*, document TS 36.300 V16.2.0, 3GPP, Jul. 2020.
- [21] *NR and NG-RAN Overall Description (Release 16)*, document TS 38.300 V16.4.0, 3GPP, Jan. 2021.
- [22] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylä-Jääski, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4150–4161, Jun. 2019.
- [23] Y. Liu, W. Wang, Y. Ma, Z. Yang, and F. Yu, "Distributed task offloading in heterogeneous vehicular crowd sensing," *Sensors*, vol. 16, no. 7, p. 1090, Jul. 2016.
- [24] H. Ko, S. Pack, and V. C. M. Leung, "Mobility-aware vehicle-to-grid control algorithm in microgrids," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2165–2174, Jul. 2018.
- [25] J. Lee, G. Lee, and S. Pack, "Pseudonyms in IPv6 ITS communications: Use of pseudonyms, performance degradation, and optimal pseudonym change," *Int. J. Distrib. Sensor Netw.*, vol. 2015, pp. 1–7, May 2015.
- [26] E. Evdokimova, A. Vinel, N. Lyamin, and D. Fiems, "Internet provisioning in VANETs: Performance modeling of drive-thru scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2801–2815, Jul. 2020.
- [27] D. Suh, H. Ko, and S. Pack, "Efficiency analysis of WiFi offloading techniques," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3813–3817, May 2016.
- [28] J. Yao and N. Ansari, "QoS-aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 167–175, Mar. 2019.



SDN/NFV, and the IoT.

**YEUNWOONG KYUNG** received the B.S. and Ph.D. degrees from the School of Electrical Engineering, Korea University, Seoul, South Korea, in 2011 and 2016, respectively. He was a Staff Engineer at the Advanced CP Laboratory, Mobile Communications Business, Samsung Electronics. He is currently an Assistant Professor with the School of Computer Engineering, Hanshin University, Osan, South Korea. His current research interests include mobility management, mobile cloud computing,

...