IoTaaS: Drone-Based Internet of Things as a Service Framework for Smart Cities

Mohammad Aminul Hoque, Mahmud Hossain, Shahid Noor, S. M. Riazul Islam, and Ragib Hasan, *Member, IEEE*

***Abstract*—The Internet of Things (IoT) offers new services in the context of smart cities through digital devices embedded with sensing, computation, and communication capabilities. The IoT devices enhance the smart city vision by employing advanced communication and computation technologies for smart city administrations. The IoT-based smart city applications require many IoT devices and gateways to be deployed at different city points. Heterogeneous sensing devices, placing smart devices in a constrained or physically inaccessible area, and large urban areas to monitor together make IoT node deployment and sensing management tasks difficult, time-consuming, and expensive. Additionally, certain tasks may require smart devices to be deployed for a very short period of time to sense and report contextual information, making it economically infeasible to purchase the devices. In this regard, we propose a drone-based IoT as a Service (IoTaaS) framework that enables the dynamic provisioning or deployment of IoT devices using drones. IoTaaS allows IoT devices and gateways to be mounted on drones and provides a distributed cloud service by placing the IoT devices in an area according to the requirements specified by a user. We also provide an economic analysis for operating such drone-based IoT services. A proof-of-concept implementation of IoTaaS for smart agriculture and air pollution monitoring applications shows that IoTaaS can reduce setup costs and increase the usage of IoT devices.**

***Index Terms*—Drones, gateway, Internet of Things (IoT), provisioning, smart city, unmanned aerial vehicles (UAVs).**

I. Introduction

THE Internet of Things (IoT) is a technology paradigm that offers intelligent connectivity among uniquely identifiable smart devices and develops a smart pervasive framework. The number of connected IoT devices is expected to be more than 40 billion by 2025 [1]. In the concept of IoT, billions of physical devices are connected through the Internet, and they are capable of collecting and sharing data. IoT devices have gained massive popularity in numerous systems, such as smart homes, smart healthcare, industrial systems, surveillance equipment, precision farming, and connected vehicles [2]–[7] where they can communicate and interact as a cyber–physical system. The idea of IoT-based smart city applications by integrating them in different city places is increasingly becoming popular [8]–[11]. For the fulfillment of the smart city vision, a considerable amount of different IoT devices are required for different sensing purposes throughout the city. However, deploying a huge amount of IoT devices is costly and time consuming. Determining the appropriate set of IoT nodes and their deployment positions imposes challenges in this regard [12]. Moreover, an IoT device needs to provide data depending on sensing time, location, and power. Hence, a framework is required to provide on-demand IoT-based services that can fulfill these requirements to enable different smart city applications.

IoT devices provide a lot of applications in the context of the smart city with varying requirements. For instance, readings from sensors may be required under different conditions and times such as environmental pollution monitoring [13]. Video surveillance [14] or vacant parking space detection [15] tasks require movement of the IoT devices. Immediate deployment may also be required in different use cases such as a disaster scenario [16], [17]. Moreover, the availability of the IoT nodes may be limited if they need to provide measurements with high precision or if they are of special uses [18]. Managing data and extracting information for better decision making are usually harder than collecting data from such heterogeneous IoT devices [19]. Considering such a wide range of sensing requirements, a dynamic IoT deployment framework can resolve the issues of selecting the correct set of IoT nodes, deployment place, and sensing management.

A drone-based distributed cloud service can enable the dynamic IoT infrastructure deployment framework by facilitating the users to rent the devices based on their requirements. Drones are feasible for this purpose because they are easy-to-deploy, capable of carrying payloads, reprogrammable during runtime, and able to measure anything from anywhere [20]. Many IoT-based applications in the smart city context leverage drones as the medium of carrying the devices to the service area [15], [17]. Fig. 1 shows several drone-based smart city applications of IoT devices. The renting mechanism of the drone-based IoT service framework allows the users to rent IoT devices at any scale and upscale or downscale later. Besides this, IoT sensing-as-a-service for different IoT use

Manuscript received August 23, 2021; revised October 31, 2021; accepted November 29, 2021. Date of publication December 30, 2021; date of current version July 7, 2022. This work was supported by the National Science Foundation under Award CNS-1351038 and Award ACI-1642078. *(Corresponding author: Ragib Hasan.)*

Mohammad Aminul Hoque and Ragib Hasan are with the Secure and Trustworthy Computing Lab, Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL 35294 USA (e-mail: mahoque@uab.edu; mahmud@uab.edu; ragib@uab.edu).

Mahmud Hossain is with the Department of Cybersecurity Engineering, Visa Inc., Foster City, CA 94404 USA (e-mail: mahhossa@visa.com).

Shahid Noor is with the Department of Computer Science, Northern Kentucky University, Highland Heights, KY 41099 USA (e-mail: noors2@nku.edu).

S. M. Riazul Islam is with the Department of Computer Science and Engineering, Sejong University, Gwangjin-gu 05006, South Korea (e-mail: riaz@sejong.ac.kr).

Digital Object Identifier 10.1109/JIOT.2021.3137362

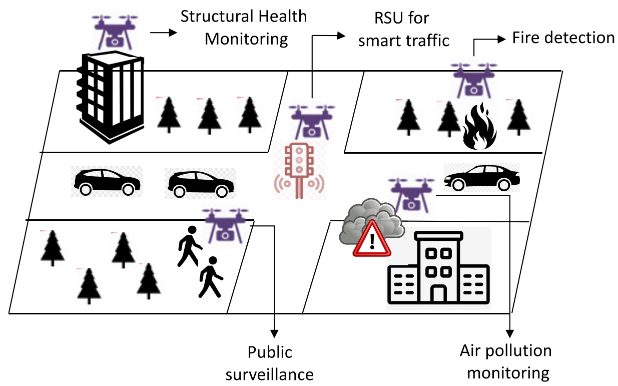

Fig. 1. Examples of drone-based IoT applications in smart city.

cases lets the users receive and interact with IoT data in a pay-as-usage model [21]. Such a model enables the service provider to generate larger profit if the IoT devices' actual usage is higher than the case of purchasing by the user [22]. The on-demand provisioning or deployment of IoT devices can reduce the setup and management issues of an IoT system and increase the usage of smart devices, hence becoming profitable for both the user and service provider.

In this article, we propose an IoT as a Service (IoTaaS) framework that provides on-demand IoT devices using drones. We assume that the drones are equipped with proper IoT devices and gateways. The users can request IoT devices and the service providers provide the devices through drones. The drones fly to the user-specified areas and create a dynamic ad-hoc IoT system. Drones are also equipped with proper communication devices to collect data from heterogeneous IoT nodes and deliver them to a cellular network. The framework requires considering the following aspects to operate smoothly: drones' short flight time, coverage area depending on the payload, optimizing the drone selection, and a proper economic model and billing mechanism for the service's sustainability. In the proposed framework, we figure out the maximum coverage area of drones by considering their short flight time. We analyze the flight time and number of required drone centers based on the payload for different setups of IoT devices. We also perform the economic analysis of the proposed framework in terms of setup cost and maintenance cost. A billing mechanism is demonstrated based on the energy consumption by the IoT devices, total service time, and the total number of transferred messages which reflect the actual usage during the service period. We demonstrate the feasibility of the framework through a proof-of-concept implementation on Contiki [23]-powered IoT devices. The experimental evaluations for smart agricultural fields and environmental air pollution show that the IoTaaS framework can reduce setup costs and increase usage of IoT devices.

Contribution: The specific contributions of this article are as follows.

1) We propose IoTaaS, a drone-based IoTaaS framework for smart cities to reduce deployment costs and ensure proper usage of IoT devices. For this purpose, we introduce a framework that enables renting and dynamic deployment of IoT devices using drones to a specific service area. The framework optimizes the drone selection and keeps track of the usage of the IoT devices conforming to the quality of service.
2) We analyze the required number of drone centers in different smart city places. For this purpose, we analyze the drone payload, flight time, and coverage area to figure out the number of required drone centers to be established.
3) We provide an economic analysis of the IoTaaS framework that includes a detailed cost analysis and a billing model to run the framework in an economically feasible manner. The study also provides cost analysis and comparison between renting and purchasing IoT devices.
4) We implement a proof of concept of the IoTaaS to demonstrate its feasibility in smart city applications. We consider smart agricultural field and air pollution monitoring applications to analyze our proposed framework.

Organization: The remainder of this article is organized as follows. Section II provides the related background and motivation to the problem. Section III explains the details of the proposed framework. Section IV provides the economic analysis for the IoT service providers. Section V provides experiment and evaluation of the framework. Section VI explores the related works of drone-based IoT services. Finally, we conclude in Section VII.

II. Background and Motivation

Smart city refers to the implementation of pervasive and ubiquitous computing in urban environments through smart technology, smart devices, and intelligent management. The vision of a smart city can be achieved using wireless networks, sensors, and digital infrastructure [24], [25]. Smart cities are characterized as locations where sensors and wireless networks enhance the efficiency, security, and sustainability of the community [26]. Smart cities must also respond intelligently to the community's needs for public safety [27]. We already observe several IoT-enabled technologies in the smart city perspective, such as environment monitoring, intelligent transport, healthcare monitoring, waste management, smart parking, pedestrian safety, etc. [10], [11], [28]–[31].

IoT devices are embedded with radio interfaces, sensors, actuators, operating systems, and lightweight services. They are capable of collecting contextual information and performing actions according to that information [29]. Several IoT devices form a group or cluster which can perform a complex task together. A cluster head synchronizes the operation of the IoT devices. The IoT devices work in a constrained and lossy network. Hence, they use several protocols, such as ZigBee [32], 6LoWPAN [33], ZWave [34], and BLE [35]. An IoT gateway bridges the communication between the outside network and IoT devices by enabling communication between heterogeneous networking protocols. For routing the packets over the network, an IPv6 routing protocol named RPL [36] is used. A typical 6LoWPAN network consists of three types of nodes: 1) leaf; 2) root; and 3) intermediate nodes, where the root node is responsible for communicating with the outside network. The 6LoWPAN nodes communicate over a wireless network defined by the IEEE 802.15.4 standard [37]. The services of IoT devices can be local or cloud. Smart devices offer local IoT services in a constrained network where the cloud IoT services let the user query a smart device from anywhere.

Unmanned aerial vehicles (UAVs) such as drones can be an integral part of establishing the smart city idea with IoT devices. They are already being used in numerous use cases ranging from commercial applications, such as product delivery [38], and governmental applications, such as power line monitoring [39]. Drones can be utilized for value-added IoT services as they are a feasible medium to carry IoT devices, sensors, and gateways to nearby distances inside the city. Drones are lightweight, economically affordable, and capable of carrying lightweight IoT devices. Due to such reasons, a drone-based dynamic IoT service platform is an appealing idea.

From user's perspective, renting IoT devices in a small scale instead of purchasing can be beneficial due to variable setup requirements and short usage time. Additionally, the users possess the flexibility to scale the IoT devices based on the requirements. Renting IoT devices through drones can also resolve the issues regarding selecting correct IoT nodes, their deployment positions, and performing optimizations depending on the IoT application [12]. Such framework deployment can also be beneficial from the service provider's perspective. The service provider of a pay-per-usage service makes higher profit when the actual usage of IoT devices is higher than the expected usage [22]. As multiple users are allowed to rent IoT devices from IoTaaS, the usage of the IoT device is expected to be higher than the fixed infrastructure setup. On the other hand, though small-scale demand may not be economically feasible for the service providers, several large users can make the service profitable. For example, Amazon Web Services (AWS) has a few large clients from whom AWS generate a huge amount of profit. The top ten clients of AWS spend U.S. $109 million monthly by using Amazon EC2 [40]. Though most of the users may demand small-scale IoT services, several large scale users, such as smart city authority, are expected to make the service profitable for the service provider. Moreover, increasing the usage of on-demand IoT services will also allow the service providers to scale the infrastructure. Hence, an on-demand IoT infrastructure service would be beneficial for both large and small-scale demands from the perspective of service provider and user. In this regard, a proper architecture and economic analysis of the framework is required.

We present the following example scenario to better motivate the drone-based IoT infrastructure deployment.

Example Scenario: Mr. X is a researcher who plans to determine the environmental pollution of the city in different places. He wants to conduct the study for a month. For this purpose, Mr. X needs a considerable amount of IoT devices to be deployed in different places. Purchasing the IoT devices for a short time is not economically feasible. Moreover, collecting data from sensors under various conditions makes the framework deployment more difficult. Mr. X can contact IoTaaS and rent the required number of IoT devices to complete the study.

IoT devices have different use cases and applications in the context of a smart city. To demonstrate the usability of IoTaaS with different applications, we analyze two categories of IoT sensor networks [12]: 1) static and 2) mobile sensor networks.

Static Sensor Network: In these networks, all the IoT nodes are placed into fixed positions. Due to their fixed locations, initial network design requires special attention in static sensor networks. Examples of such use cases include environment monitoring [41], structural health monitoring [42], etc. Such networks incur a generic problem of deciding where to deploy the nodes for better monitoring [43]. Incorporating different IoT node placement mechanisms [44], [45] can be performed by renting drone-mounted IoT nodes through IoTaaS to achieve better performance.

Mobile Sensor Network: In these networks, the sensor nodes themselves may move for collecting data based on previous measurements or tasks, or the data collector nodes may move to collect data from different sensor nodes [12], [46]. Mobile nodes help to improve network performance in terms of coverage [47], connectivity [48], and energy consumption [49]. IoTaaS framework can also serve for the use cases of mobile sensor node deployment using drones with the flexibility to execute route planning for mobile nodes [50], optimizing the drone positions [51], etc.

III. Proposed Framework

In this section, we provide the details of the proposed IoTaaS framework. We subsequently explain the desired properties of the integrated drone-based IoT framework, drone setup, and detailed architecture of the proposed framework. After that, we analyze the drone payload and flight time based on the drone setup. Finally, we explain the operation model for requirement submission and drone assignment for a service.

A. Desired Properties of the Framework

An integrated IoT service provisioning framework using drones should possess several properties. Previous works did not consider all these properties together for integrated drone-based IoT service provisioning. The desired properties are as follows.

1) Users should be able to choose proper IoT devices according to the requirements.
2) The framework should monitor each of the ongoing services and upscale or downscale based on the requirements or performance.
3) There should be a proper economic analysis and corresponding billing mechanism that would reflect the devices' actual usage.
4) The framework should be able to execute IoT device placement algorithms for better sensing results and also store service details for future auditing purpose.
5) The framework should assign drones based on the mounted IoT devices, payload, and distance by optimizing the service time or energy consumption.
6) Drones should be equipped with proper IoT devices, communication devices, and gateways. In this regard, the

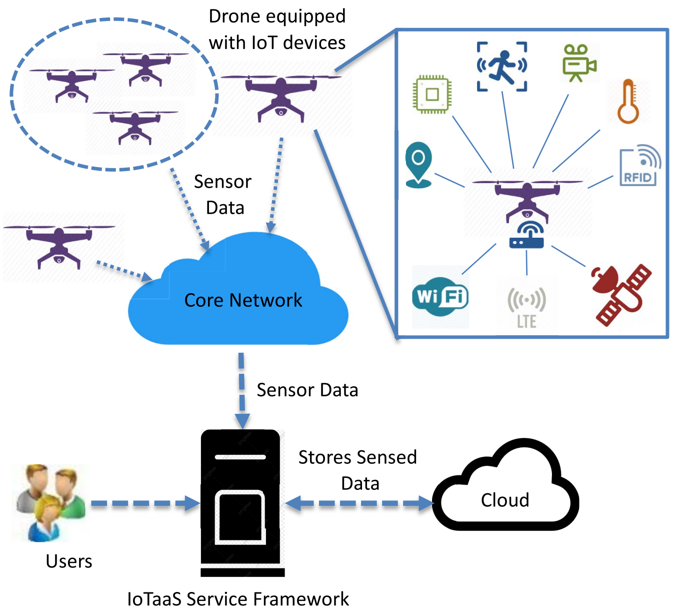

Fig. 2. Overview of the IoTaas framework.

framework should consider payload carrying capability and maximum flight time of the drone.

B. Drone Setup

Fig. 2 shows the overview of the IoTaaS framework with a drone equipped with IoT devices and gateways. The service provider establishes several drone centers in different city areas where different IoT devices and gateways are mounted on the drones to deliver the on-demand dynamic IoT infrastructure. Examples of attached IoT devices include temperature sensors, Radio Frequency Identification (RFID) devices, cameras, motion sensors, and GPS. The service provider supplies its pricing model to the IoTaaS framework based on energy consumption, service time, number of messages transferred from IoT to cloud, and user requirements. The drones are equipped with Wireless Local Area Network (WLAN—IEEE 802.11), IoT (IEEE 802.15.4), and 4G/5G network interfaces to maintain excellent and stable connectivity. It is thus possible to preprogram the drones regarding the service and send updated instructions. In addition, the drones are capable of storing the sensed data in the cloud using the mounted IoT gateway. Section III-D explains the drone payload and flight time calculation process.

C. IoTaaS System Framework

The IoTaaS framework provides a front-end for the users to provide the requirements and find out the optimized resources. Fig. 3 provides the detailed architecture of IoTaaS system framework. The framework components are as follows.

1) IoTaaS Service Framework: The IoTaaS service framework is responsible for executing and monitoring each of the concurrently running drone-based IoT services separately. The framework keeps track of the current status of each services and aggregates the available and free resources of each drone center. The system framework has the following modules.

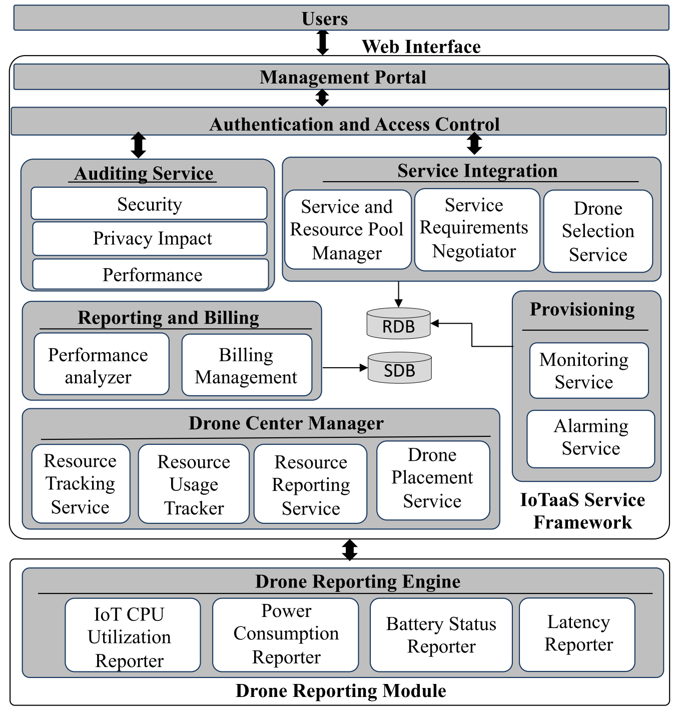

Fig. 3. System framework of IoTaaS.

Management Portal: Unlike many previous frameworks, IoTaaS contains a Management Portal that provides a Web interface for a user to explore the available resources. A user can navigate the available services and price packages through the user interface of the management portal in the cloud by providing service requirements. The user can explore the software and hardware specification details of the sensors, actuators, gateways, and topology, such as mesh, star, and tree. The user can also monitor the ongoing service and interact with the data from the cloud reported by the IoT devices. The service provider can update the resource pool and obtain information regarding ongoing services through management portal. Section III-E explains the operation model for requirement submission and drone assignment process.

Service Integration: The service integration module is responsible for analyzing the available resources to figure out the best service deals. It analyzes the service requirements and discovers the compatible resources from resource database (RDB).

The service and resource pool manager maintains the details of all the drones and mounted IoT devices in RDB. This component also keeps track of ongoing services in service database (SDB) and maintains the current information regarding the drones, such as its battery status, location, route, and installed IoT equipment.

The service requirement negotiator finds the best resource deals for a specific requirement after getting the detailed specification of the resources. Finally, the specifications are delivered to the users through management portal. This component also stores the service information in the SDB once a service starts.

The drone selection service selects the drones that can perform tasks specified by the user. The optimized drones can be selected from multiple drone centers depending on different conditions, such as distance from service place, availability of drones, availability of IoT devices, etc. A service provider may have different types of drones with varying properties, such as weight, payload capabilities, and battery. Drone selection is optimized based on service time or energy consumption which can be divided into three phases: 1) flying to the service area; 2) processing the sensed data; and 3) transmitting the data to the cloud server [52]. If the service provider has a total of n drones in multiple drone centers, then

$$E_j = E_{\text{fly}} + E_{\text{process}} + E_{\text{transmit}} \tag{1}$$

$$T_j = T_{\text{fly}} + T_{\text{process}} + T_{\text{transmit}}. \tag{2}$$

Here, E_j and T_j denotes the energy and time required to provide the service using the jth drone where $0 \leq j < n$. E_{fly}, E_{process}, and E_{transmit} denotes energy consumption for traveling, processing data, and sending the data to the cloud server, respectively. Similarly, T_{fly}, T_{process}, and T_{transmit} stand for flying time, data processing time, and network latency for sending data to the cloud server. We can define the overhead cost in terms of energy and time as follows:

$$Z_{ij} = E_j \times p_i + T_j \times q_i. \tag{3}$$

Here, p_i and q_i denote the importance weights of energy consumption and time for a specific service i where $0 \leq p_i, q_i \leq 1$ and $p + q = 1$. The weights influence the drone selection and service strategy. For example, more drones in the service area reduce the time required for the service. However, the total amount of energy consumption is increased if more drones fly to the service area. Hence, q_i will be larger than p_i for optimizing service time by putting less importance on energy consumption. Conversely, low importance on service delay with a higher span of service time with optimized energy consumption requirement puts a larger value to p_i. Hence, if there are n available drones and m service requests, we can formulate the following optimization problem:

$$\text{minimize} \sum_{j=1}^{n} \sum_{i=1}^{m} Z_{ij} \times x_{ij} \tag{4}$$

such that

$$\forall j \epsilon n : E_j \times x_{ij} \leq E_i \tag{5}$$

$$\forall j \epsilon n : T_j \times x_{ij} \leq T_i \tag{6}$$

$$\forall j \epsilon n : \sum_{i=1}^{m} x_{ij} \leq 1 \tag{7}$$

$$x_{ij} \epsilon \{0, 1\}. \tag{8}$$

Here, x_{ij} in (4) denotes a decision variable that decides whether the drone j is selected to provide service i. As the drone's movement consumes most of the energy, the energy consumption optimization leads to completing the task with the fewest number of drones and minimum movement inside the service area. In contrast, time optimization requires to select more drones with higher CPU and bandwidth capability. However, irrespective of the value of p_i and q_i, each drone j requires to maintain a threshold of energy consumption and time for providing a service i which are denoted by E_i and T_i [constraints (5) and (6)]. Thus, time and energy consumption should not exceed the threshold to maintain the quality of service. Finally, constraints (7) and (8) ensure that each drone is assigned to at most one service.

Drone Center Manager: The drone center manager is responsible for controlling one drone center. This module manages all the current services, status of the drones, provides available resource list according to the requirements, tracks current resource usage, and places drones to optimized locations.

The resource-tracking service keeps track of the drones of a drone center and reports to service and resource pool manager to update the available resource list in RDB. The resource-tracking service also provides the drone center's compatible and available resources according to the requirements received from the service integration module.

The resource usage tracker (RUT) tracks the resource utilization in terms of power consumption, memory, bandwidth, and CPU utilization of the IoT devices. Resource usage tracking is essential for billing purposes at the end of service.

Resource reporting service reports the service details periodically to the alarming module of provisioning service to ensure the quality of service. This component also updates the resource usage summary in SDB at the end of the service.

Drone placement service is responsible for placing the drones to optimized locations for maximizing the sensing output. For this purpose, the drone placement service executes related IoT node placement algorithms [44], [45].

Provisioning Service: The provisioning service enables upscaling and downscaling the resources. For this purpose, this service monitors the resource usage and verifies the service-level agreement (SLA) and Quality of Service. A user can specify usage threshold parameters for the allocated services and define the required actions to be performed when a metric exceeds the threshold. The monitoring service can also generate log provenance for forensic investigation and performance audit of the service provider in future. The monitoring service receives usage reports from resource reporting service and monitors whether they are performing as expected. Alarming service informs the drone center manager regarding any unsatisfactory condition in sensing or battery status to provision with more drones.

Auditing Service: The auditing service is composed of three components: 1) security audit module; 2) privacy impact audit module; and 3) performance audit module. The auditing module periodically logs the activities among the users, IoTaaS service provider, and the IoTaaS Service Framework. This module is also responsible for maintaining the secure provenance of the logs for post-verifiable auditing. The auditing service provides an interface for auditors to retrieve and analyze the stored logs for forensic investigation. Previous drone-based IoT frameworks [14], [53], [54] do not support auditing services, which is crucial for future investigation of a service-related incident.

Reporting and Billing: The performance analyzer module generates reports based on service usage and SLAs from SDB. Then, the module calculates provides the service report to billing management component. The billing management

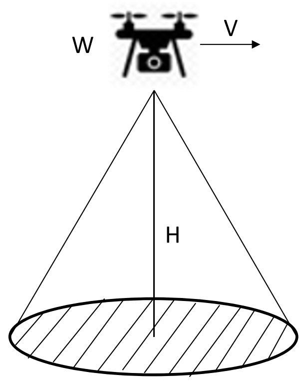

Fig. 4. Parameters that affect drone flight time and coverage area: total weight with payload (W), height from the ground (H), and velocity (V).

component considers all the usage report to generate the bill. Section IV provides economic analysis and billing mechanisms that reflect the actual usage of the rented devices. Though several works [55], [56] proposed billing mechanisms for the IoT device renting framework, they do not provide complete economic analysis in this regard.

2) Drone Reporting Engine: The drone reporting engine is responsible for reporting the current status of the service periodically to the IoTaaS service framework. The reporting module includes the CPU utilization reporter, power consumption reporter, drone battery status reporter, and latency reporter. These reporters are required for the service to maintain the quality of service and keep track of the usage to bill the user later. A Raspberry Pi that works as the gateway in the IoT networks is responsible for executing these modules. Here, we explain the responsibilities of different reporting modules.

CPU Utilization Reporter: CPU utilization refers to different utilization metrics of IoT devices, such as RAM, bandwidth, CPU cycles, etc. CPU utilization reporter periodically checks these metrics and sends report through drone reporting engine.

Power Consumption Reporter: This component is responsible for keeping track of consumed energy while transferring data between the IoT device and the cloud server.

Battery Status Reporter: As the IoT devices consume energy from the drone battery, it is crucial to monitor the drone battery status continually. The battery status reporter module monitors the drone battery to determine the drone's return time to ensure the safe return to the drone center.

Latency Reporter: In the IoTaaS architecture, drones can collect data individually or multiple drones can create an ad-hoc IoT cloud. The latency reporter module sends ping messages to different IoT devices and calculates the round trip time that represents the latency.

D. Drone Payload and Flight Time Calculation

The drones must have enough energy before being assigned for a service task. Here, we determine the payload weight of IoT devices and gateways the drone can carry and the drone's maximum flight time with that payload. Fig. 4 shows the parameters that affect drone flight time and coverage area. Here we explain the parameters briefly.

Flying Position and Speed: The requirements of flying to the service area and maintaining a height from the ground affect the total energy consumed by a drone. Placing a drone to a much higher position requires the drone to generate a much higher amount of thrust and significantly larger energy. Moreover, the hover position draws much less energy than moving forward at a certain speed. Again, more speed and wind from the opposite direction also draw more energy.

Weight: The energy consumption for flying a drone is also proportional to the amount of weight carried by a drone. Besides carrying the IoT devices, a drone itself has some weights, such as frame, motors, electronics, battery, accessories, etc.

Ideally, we need to maintain around thrust-to-weight ratio of 2:1. Suppose the weight of a drone without considering the weight of battery and IoT devices is w_d. The average weight of battery and each IoT device are w_{bt} and w_{iot}, respectively. Drone motors can generate w_{th} maximum thrust on an average. Hence, the maximum number of IoT devices a drone can carry is

$$n_{\text{iot}} = \frac{\frac{w_{\text{th}}}{2} - w_d - w_{bt}}{w_{\text{iot}}}. \tag{9}$$

Now, we determine how long we can use a fully charged drone with payload before we need to call them back for recharging. The flight time of a drone is calculated as

$$\text{Flight time} = \text{capacity} * \text{discharge}/\text{AAD}.$$

Here, the capacity refers to how much current the battery can hold when it is fully charged and the discharge indicates the maximum allowed level to discharge the drone battery. Usually, the recommended level of battery discharge is 80% to ensure the good health of the battery. The average Ampere draw (AAD) can be calculated as follows:

$$\text{AAD} = \text{AUW} * P/V.$$

All up weight (AUW) refers to the take-off weight, P denotes the power required to lift 1 kg weight, and V stands for the voltage of the drone battery. P varies with the drone speed as low speed costs low power and high speed draws more power. However, an optimal speed provides the maximum efficiency to a drone that depends on the drone weight and battery. Maximum efficiency of the drone allows the drone to cover maximum path.

E. Operation Model

Based on the requirements provided by the user, the IoTaaS framework figures out the optimized drones and assigns for the service. Fig. 5 provides the operation model for requirement submission and drone assignment process. Here, we explain all the steps of the operation model.

Step 1: The users provide requirement specification through the management portal. *Steps 2* and *3:* The requirements are forwarded to the resource pool service. *Step 4:* The requirements are analyzed to define the specification from the

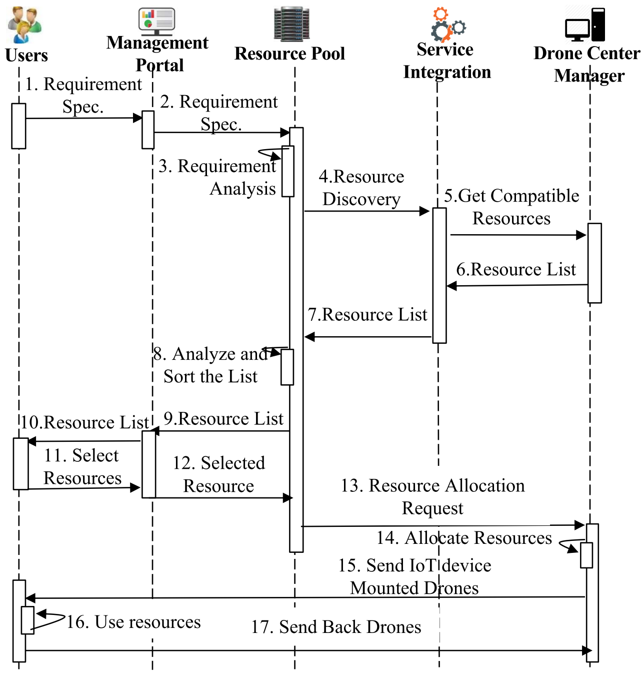

Fig. 5. Operation model of IoTaaS framework.

resource pool specifications. *Step 5:* Requirement specification is provided to the drone center manager for getting the compatible resource list. *Step 6:* The drone center manager returns the resource list that fulfills the requirement specification. *Steps 7 and 8:* The resource list is returned to the resource pool and the list is sorted to find the best options at the start. *Steps 9* and *10:* The list is returned to the user through the management portal. *Step 11:* The user selects the most feasible option from the list. *Step 12* and *13:* The selected option is forwarded to the resource pool and the resource allocation request is sent to the corresponding drone center. *Step 14:* The drone center allocates necessary resources to the user. *Step 15:* The drone center manager sends the IoT device and gateway mounted drones to the specific site defined by the user. *Step 16:* The user uses the resources to perform the specific sensing task. *Step 17:* At the end of the resource usage, the drones are sent back to the drone center.

IV. Economic Analysis

In this section, we provide an economic analysis of the IoTaaS framework. The cost model derived from the economic analysis can be used to estimate the financial feasibility of IoTaaS-based service.

A. Economic Analysis of IoTaaS

To perform the economic analysis of IoTaaS, we provide a cost model for setting up and running the IoTaaS service. Then, we propose a billing mechanism to use IoTaaS based on two strategies: 1) total consumed energy and 2) the total number of transferred messages during service time.

1) Setup Cost: Initial costs for system setup include purchasing the service equipment, renting places to operate the service, and setting up the system framework. The service provider needs to establish several drone centers in different city areas. To provide IoT infrastructure services, they need to purchase drones, IoT devices, IoT gateways, and other auxiliary equipment. The drone centers contain charging stations to recharge the drone batteries. Some hardware and software are also required to set up the system framework and device maintenance. Service providers usually hire experienced people or third-party vendors for the initial setup. Moreover, the service providers need to rent virtual machines to host the cloud service. So, the initial setup cost would be

$$\text{Initial cost} = \text{device purchase cost} + \text{infrastructure setup cost}.$$

If the service provider wants to cover a smart city area and ensure the drones will spend at most $x\%$ of their energy for providing service, then the maximum fly time of a drone t_s can be calculated using the properties explained in Section IV. If the drones have an average speed v, then we can calculate the maximum radius of the area they can cover according to the following formula:

$$r_{\max} = \frac{x}{2 * 100} \times v \times m. \tag{10}$$

To reduce the setup cost, the service provider wants to set up as few drone centers as possible. Hence, they need to minimize the overlapping of the coverage area. Instead of the circular shape of the coverage area, we are considering the shape as square. Hence, the maximum area that one drone center can cover is

$$\text{area}_{\max} = 2 \times r_{\max}^2. \tag{11}$$

As we have assumed the coverage area is a, the minimum amount of drone centers need to be established in the smart city will be

$$n_{\text{droneCenter}} = \frac{a}{\text{area}_{\max}}.$$

If the service provider targets to serve y users at a time with p drones to each user, then the total number of drones (n_{drones}) required for all drone centers will be

$$n_{\text{drones}} = n_{\text{droneCenter}} \times y \times p. \tag{12}$$

Hence, the purchasing cost for drones will be

$$n_{\text{drones}} \times \text{drone}_{\text{unitPrice}}. \tag{13}$$

2) Maintenance Cost: The devices used for providing service require proper maintenance to ensure the quality of service. Hence, the service provider needs to hire some regular employees to ensure appropriate device maintenance. Moreover, the provider has to pay the rent of drone center spaces established in different smart city locations. The total maintenance cost can be calculated by aggregating all these expenses.

The cost of repairing devices can be calculated using the probability of a device's failure in a day and the average cost for fixing such a device. Multiplication by the total number of failed devices provides the total repair cost:

$$\begin{aligned}\text{Repair Cost} = {} & \text{Failing probability (in a day)} \times \text{No of failed} \\ & \text{device} \times \text{Average cost for repairing the device.}\end{aligned}$$

The cost for employees can be calculated as

$$\text{Employee cost} = \text{Total no. of employee} \times \text{Average salary of an employee.}$$

Rental cost for drone centers is another important cost for the service providers. The rental cost can be calculated as

$$\text{Rental cost} = \text{Total number of drone centers} \times \text{Average rent of drone centers.}$$

The power consumption cost is incurred by the drones and other hardware and networking devices. We can calculate the power consumption cost as

$$\text{Power consumption cost} = \text{Number of device} \times \text{Average Energy Consumption} \times \text{Unit Energy Cost.}$$

The total maintenance cost can be calculated by aggregating all the costs.

$$\text{Maintenance cost} = \text{Repair cost} + \text{Employee cost} + \text{Rental cost} + \text{Power consumption cost.}$$

3) Billing: As our proposed framework works in a pay-as-you-go model similar to the cloud, the billing mechanism depends on the usage of the rented resources. We propose the billing method based on the total number of messages sent to the cloud server and total consumed energy for sensing and reporting. We also consider the dynamics of IoT traffic and incorporate that into the billing mechanism. Here we illustrate the entire billing method along with the traffic modeling.

Dynamics of IoT Traffic: The fundamental theorem for traffic processes is the Palm–Khintchine theorem which states that large number of independent traffic process can be described by a Poisson process. IoT traffic from multiple nodes can also be described using the theorem. For the Poisson process, inter-arrival time X follow an exponential distribution with traffic arrival rate λ with probability density function [57]

$$X \sim \text{Exp}(\lambda) : f_x(t) = \lambda e^{-\lambda t}. \tag{14}$$

Hence, the expected interarrival time for a traffic process i would be

$$E[X_i] = \frac{1}{\lambda_i}. \tag{15}$$

Here, λ_i is the traffic arrival intensity of process i. Let us assume that there are n IoT devices attached to the drone. If all the nodes send data at a similar periodic interval T, then approximation of the Poisson process provides the expectation of interarrival time as follows:

$$E[X_i] = \frac{T}{(n+1)}. \tag{16}$$

However, in our proposed framework, there can be multiple IoT nodes with different periods as the IoT devices may be used for different purposes. Moreover, some IoT devices can be event driven instead of reporting data periodically. Fig. 6 shows a example of traffic arrival patterns from different IoT nodes mounted in a drone. Hence, we model a mixed traffic scenario that aggregates both the Poisson traffic and periodic traffic. To model such heterogeneous traffic, we consider α fraction of total IoT traffic arrive as the Poisson process and the remaining $1-\alpha$ fraction arrive as the periodic process. Here, all the Poisson and periodic traffic are aggregated separately to figure out the total number of messages reported by all the IoT devices mounted in the drone.

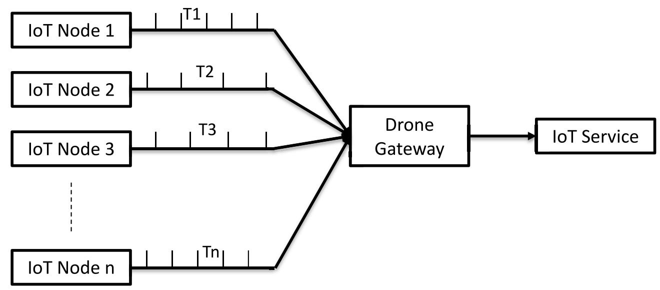

Fig. 6. Arrival of IoT traffic.

Billing Based on Total Consumed Energy: The total bill is calculated based on consumed energy by all the IoT devices operated in the drone. We consider the following parameters while service being provided by a single drone.

1) Consumed energy for reporting each unit of data is E_r.
2) The size of each data reported by an IoT device is x.
3) The total usage time of drone-based service is m.

Energy consumption-based billing mechanism leverages the message arrival rate. Higher message rate (low inter arrival time) incurs more bill and vice versa. If the total number of messages reported by the IoT devices is r, observed inter arrival time is t, and expected inter arrival time is t_{exp}, then the energy consumption by all IoT devices of a drone will be

$$E_{\text{drone}} = \frac{(E_r \times x \times r)}{\left(t/t_{\text{exp}}\right)}. \tag{17}$$

Let us assume the cost for consuming each unit energy is c. Moreover, the bill also depends on total service time. If a drone is rented for m minutes and cost for each minute is z, then we can calculate cost for one drone as follows:

$$C_{\text{drone}} = (E_{\text{drone}} \times c) + (m \times z). \tag{18}$$

If the drone's performance point is d and the provider wants $p\%$ profit, then the billing amount for one drone will be

$$B_{\text{drone}} = (1 + p/100) \times C_{\text{drone}} \times d. \tag{19}$$

The profit margin covers all other costs and profits of the service provider. The total bill for using n drones for the service will be calculated as

$$B_{\text{total}} = \sum_{n=1}^{n} B_{\text{drone}}. \tag{20}$$

Billing Based on Total Number of Transferred Messages: In this method, the bill is calculated based on the total number of transferred messages between the IoT devices and the cloud server. The framework considers the maximum size of a message as 1 kB. Hence, a message of size more than 1 kB is divided into multiple messages. If the average message size is x bytes and all the IoT devices of a single drone send p

messages, then the total number of messages calculated by the billing mechanism is

$$M_{\text{drone}} = (x/1024) \times p. \tag{21}$$

If the cost for per message is c, and the drone is rented for m units time in a rate of z cents, then the cost for a single drone can be calculated as follows:

$$C_{\text{drone}} = (M_{\text{drone}} \times c) + (m \times z). \tag{22}$$

After calculating the cost for a single drone, the total billing amount can be calculated using the equations 16 and 17. For both the billing strategy, the service provider can adjust the per unit price (based on consumed energy or message) and profit margin by maintaining the SLA.

B. Economic Analysis of Fixed Infrastructure

We explain the economic analysis of the fixed infrastructure by analyzing device purchase cost and maintenance cost. Device purchasing costs for the fixed infrastructure include IoT device purchase costs and drone purchase costs. Purchased devices need to be adequately maintained, which incurs device repair costs, rental costs, and power consumption costs. All these costs are identical to the setup and maintenance costs of the IoTaaS architecture. However, the costs will depend on the application scale, which will vary based on different use cases of the IoT devices.

For economic analysis, we use the concept of net present value (NPV), which is popularly used to calculate an investment decision's profitability. Walker [58] used this methodology to explore the feasibility of renting computation resources from the cloud. Calculating NPV is important as it considers all cash inflows and outflows during the lifetime of the investment. Moreover, it also considers the decrements of dollar value over time. We adopt the methodology for calculating NPV from [59] which is defined as

$$\text{NPV} = \sum_{t=0}^{Y-1} \frac{C_t}{(1+r)^t}. \tag{23}$$

Here, Y denotes the number of years, r is the rate at which the dollar's value decreases, and C_t is the cost at time t.

Time Value of the IoT Devices: Now, we analyze the cost per hour for the purchase case (setting up the fixed infrastructure) and lease case of IoT devices. For this purpose, we adopt the analysis mechanism from [58]. According to Moore's law, the number of integrated circuit transistors is expected to be doubled every two years, leading the CPU capacity to be doubled in that timeframe. We assume the law is also applicable for IoT devices. Hence, the cost per hour for both the fixed infrastructure and lease case would be

$$\text{cost} = \frac{\text{NPV}}{\text{NPC}}. \tag{24}$$

Here, NPV is the net present value, and NPC represents the net present capacity, which is the total hours of usage (CPU hours) of all the considered IoT devices. The total capacity (TC) is calculated using the total working hours for a specific period. For example, if an IoT device works 50% of its total capable time, the TC for one year would be 365 × 24 × 0.5 h. However, due to CPU depreciation according to Moore's law, the NPC for next Y years would be

$$\text{NPC} = \text{TC} \times \frac{1 - \left(\frac{1}{\sqrt{2}}\right)^Y}{1 - \frac{1}{\sqrt{2}}}. \tag{25}$$

Hence, the cost per hour for fixed infrastructure case would be [from (24)]

$$\text{cost} = \frac{\left(1 - \frac{1}{\sqrt{2}}\right) \times \left(\sum_{t=0}^{Y-1} \frac{C_t}{(1+r)^t}\right)}{1 - \left(\frac{1}{\sqrt{2}}\right)^Y \times \text{TC}}. \tag{26}$$

We assume that the ervice provider updates the IoT devices with time. Hence, the user does not face the CPU depreciation cost. For renting IoT devices, TC is the total usage hours of the IoT devices. Hence, total cost per hour for renting devices would be

$$\text{cost} = \frac{\sum_{t=0}^{Y-1} \frac{C_t}{(1+r)^t}}{Y \times \text{TC}}. \tag{27}$$

V. Experiment and Evaluation

In this section, we explain our conducted experiments and evaluations to demonstrate the feasibility of the proposed framework. We initially calculated the drone coverage area and the number of required drone centers considering different sensor setups. We further considered the smart farming scenario to demonstrate the renting cost compared to the case of purchasing the devices. Moreover, we present the cost comparison of investment alternatives for several years. Finally, we considered the industry air pollution monitoring scenario to find the usage of the sensors in different circumstances.

A. Number of Drone Centers

We considered DJI Phantom 2 drones for providing the IoT services. The weight of the DJI Phantom 2 drone is 1000 g. These drones are capable of lifting up to 1300 grams of weight during take-off. The drone has a 5200-mAh battery with 25 min of hover flight time. The drone draws approximately 290-W power to lift 1 kg weight while operating at 40 km/h, which optimizes the traveled distance and power consumption. As the drone can fly with a maximum of 300 g of extra weight, it is impossible to mount all the sensors for different purposes in one drone. For the purpose of our experiment, we considered two different drone setups to use for various IoT applications. The purpose of one setup was precision farming, where the drones were equipped with a salinity sensor, moisture sensor, gas sensor, weather station sensor, and Raspberry pi. The total weight of the drone became 1175 grams after mounting all the sensors in the drone. The other setup was used for disaster management and pollution monitoring that included Bluetooth beacon, GPS device, smoke detector, humidity sensor, air quality sensor, and Raspberry pi having a total weight of 1275 g.

For drone center setup, we considered a smart city with an area of 400 km^2 and calculated the approximate number of drone centers required to cover the city area. We assumed that drones are assigned to cover the area under a particular drone center. Fig. 7 shows the coverage area of a drone considering

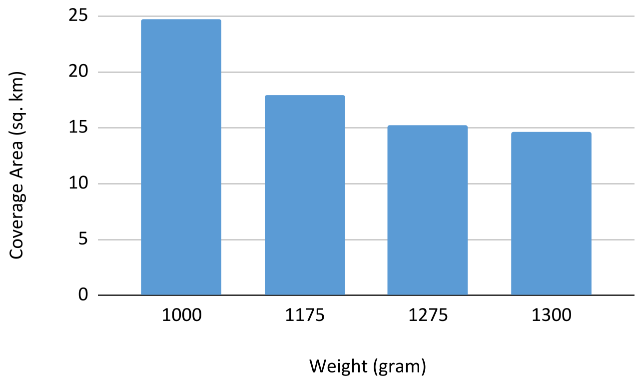

Fig. 7. Coverage area of drone with different weight.

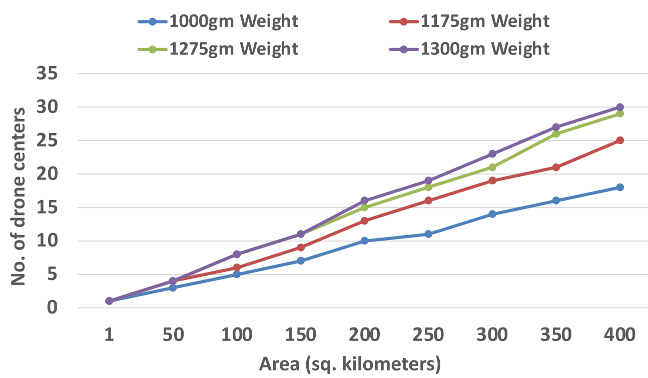

Fig. 8. Number of drone centers required for different payloads in drone.

the take-off weight. The coverage area was calculated using the drone flight time calculation formula explained in Section IV. Fig. 8 shows the number of drone centers required to cover the city area. During the calculation, we considered that the drones would spend at most their 75% of battery life, including returning to the drone center.

B. Energy Consumption Based on Optimization Constraints

We evaluated the effects of optimization constraints described in Section IV by implementing the solution of the optimization problem in python and varying the service time and energy consumption constraints. The importance weights of energy consumption and service time (p and q, respectively) resulted in different energy consumption levels. The simulation was performed for a variable number of drones and corresponding energy consumption depending on the constraints. We varied the total number of drones from 50 to 500 and calculated the energy consumption considering they will maintain around 25% of their battery level after returning to the drone center. Fig. 9 shows total energy consumption with variable values of importance weights. We observed that higher importance on service time ($q = 0.9$) caused high energy consumption with an increasing number of drones. This approach decided to incorporate more drones to reduce service time without considering total energy consumption. For equal weights on time and energy consumption constraints ($p = 0.5$ and $q = 0.5$) shows less energy consumption than the previous approach. In this case, the algorithm tried to assign a limited number of drones to reduce energy consumption and maintain a reasonable service time duration. Finally, more weights on energy consumption

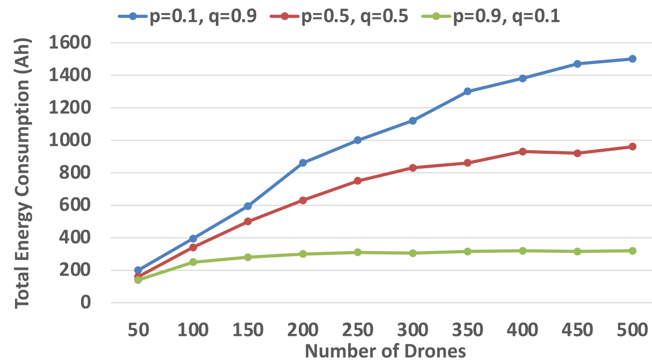

Fig. 9. Energy consumption for different number of available drones with variable optimization constraints.

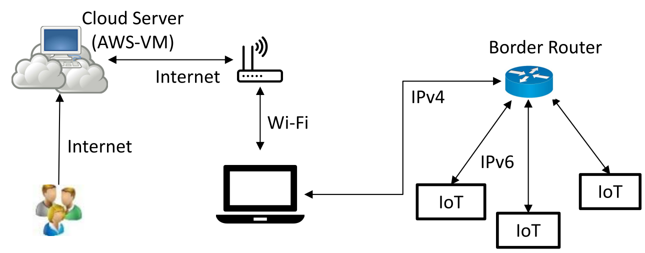

Fig. 10. Experimental network setup.

optimization ($p = 0.9$) reduced total energy consumption by selecting fewer drones, which introduced a delay in service.

C. Smart Agriculture Field

We considered a smart Agriculture [60] scenario for our experiment where IoT devices measure the environmental context of the agriculture field and report them to keep the farmers informed about the current condition. We considered usage of different sensors for smart farming, such as salinity, moisture, weather station, and location sensors. To calculate the number of sensors required for smart farming, we considered an acre of farming land and divided it into multiple blocks according to the signal strength of the sensors' radio transceiver. The drones were equipped with all four sensors and a gateway capable of communicating with both IPv4 and IPv6 protocols. The IEEE 802.15.4 has around 10-m communication range. Hence, the sensor can cover approximately 100 m^2 of farming land. A total number of 41 drones were required to cover the full 1 acre of farming land. We assumed that the drones will not move inside the farming land which lead to the requirements of 41 drones to cover full one acre land. However, drones can be programmed to move around the field and collect data to reduce the total number of drones required.

1) Experimental Setup: We created a proof of concept implementation of IoTaaS that runs on RE-Mote IoT device powered by the Contikiti [23] operating system. The device was simulated using the Cooja simulator [61] and operated at 8-MHz CPU speed, 64-kB RAM, and 512-kB ROM. The IoTaaS service framework was running as a Web Service on a Virtual Machine located in Amazon Cloud. We performed the simulation in a MacBook Pro with a Core i5 processor and 8-GB RAM. We created a gateway application that was running on the laptop. Fig. 10 shows our experimental

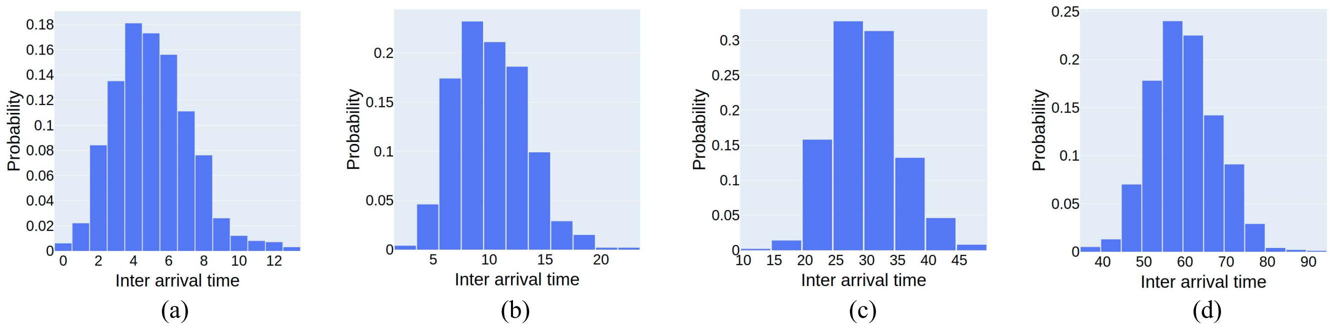

Fig. 11. Probability distribution of inter arrival time (in second) of data for (a) moisture sensor, (b) salinity sensor, (c) location sensor, and (d) weather station sensor.

network setup for one drone where the IoT devices created a star topology.

2) Evaluation: We performed an analysis of the effectiveness of our proposed solution in terms of the cost of purchasing individual sensors and renting sensors embedded in a drone. For the fixed infrastructure with purchased IoT devices, we calculated device purchasing cost and their operating cost. Purchasing costs for all the sensors for an acre of land would be approximately U.S. \$3362 based on the current market price.

We calculated the energy consumption for transferring sensed data for each IoT device to determine the incurred bill in IoTaaS. For this purpose, we calculated the message size for each request sent to the cloud server from each sensor mounted in the drone. The IoT devices sent the data using post request, and the payload size was $x1 = 316$ bytes for the moisture sensor, $x2 = 309$ bytes for the sanity sensor, $x3 = 535$ bytes for the weather station, and $x4 = 375$ bytes for the location sensor. As an explanation of the message sizes, the JSON object generated by the weather station sensor contains the following information: temperature in Fahrenheit, pressure in pascal, humidity in percentage, dew point temperature in Fahrenheit, wind speed in meter/second (m/s), the maximum speed of wind gusts in m/s, and rainfall value in millimeter.

We assumed that the moisture sensor sends data once in every 5 s, sanity sensor sends data in every 10 s, location sensor sends data in every 30 s, and weather station sensor sends data in every 60 s. We performed simulation for data arrival intensity λ_i for all the sensors and calculated the probability distribution of inter arrival times. Fig. 11 shows the probability distribution of inter arrival times for all the sensors. Based on the inter arrival time probability distribution for all four sensors, we measured total consumed energy and calculated total cost for renting device from IoTaaS. Figs. 12 and 13 show a comparison between fixed infrastructure cost and the cost through IoTaaS framework. We presented the result with variable profit margin and price per unit energy consumption. The cost with the variable profit margin was calculated by setting the cost for per joule energy consumption as 1.5 cents. On the other hand, for the variable price per unit energy, the profit margin was fixed to 20%. The profit margin can be adjusted based on other costs of the service provider, such as drone center rent, personnel cost, etc. Cost calculation was shown for four months, considering the required time for a full

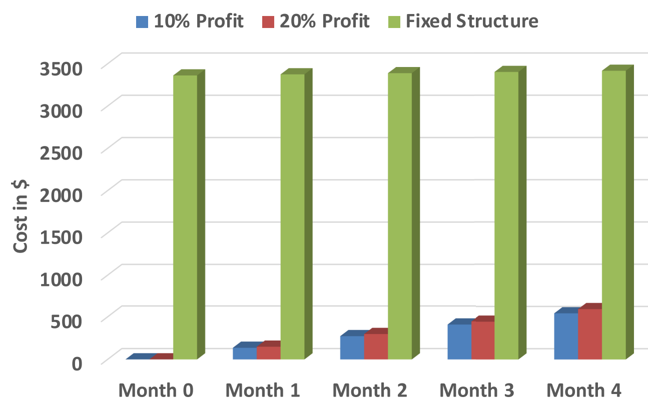

Fig. 12. Comparison of cost with variable profit margin.

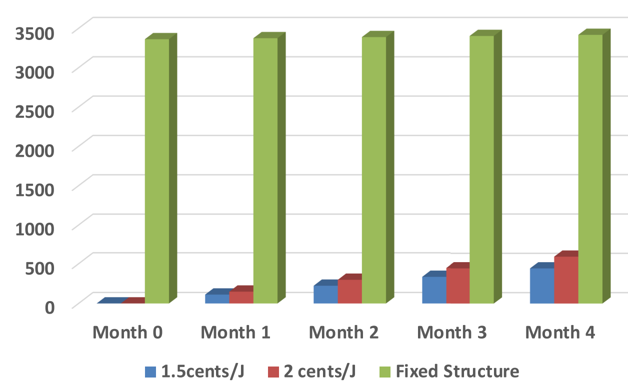

Fig. 13. Comparison of cost with variable price per unit energy for energy consumption-based billing strategy.

cycle of growing and harvesting crops. We also calculated cost considering the number of messages transferred to the cloud server. The maximum message size was 535 bytes for the stated scenario, while the maximum allowed message size was 1024 bytes. Hence, with each data report, the number of messages transferred was increased by one. Fig. 14 shows the cost with fixed infrastructure and variable price per 1000 messages. We observe that the cost is significantly lower and feasible if the sensors are rented through the IoTaaS framework.

Comparing Investment Alternatives: We calculated the cost per hour for renting and purchasing IoT devices to compare the investment alternatives. Fig. 15 shows the cost per hour comparison if the user purchases the devices and rents through the IoTaaS. Even if the user sets up his own infrastructure with

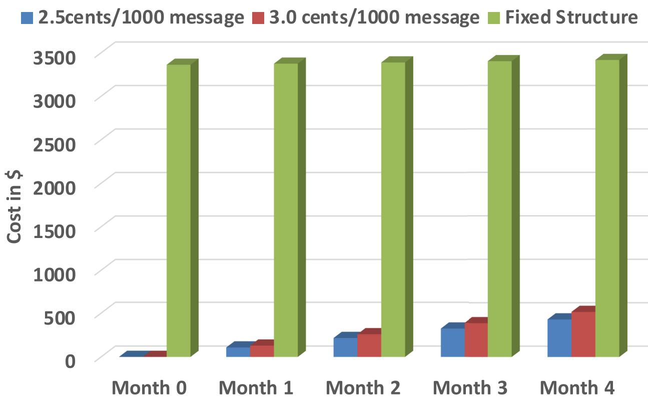

Fig. 14. Comparison of cost with variable price for per 1000 messages.

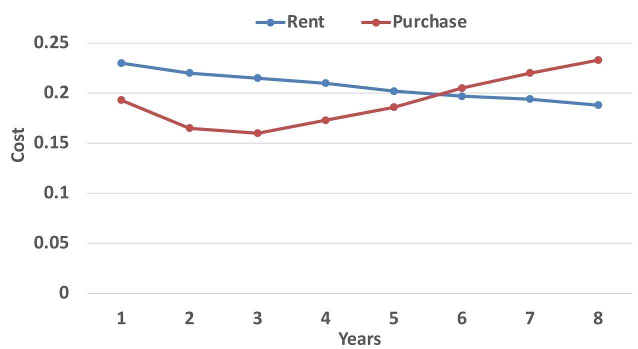

Fig. 15. Cost per hour (in dollars) comparison between renting and purchasing the IoT devices.

all the IoT devices required for one acre of agriculture field, the cost per hour will surpass renting after around five years. We performed the calculation based on the NPV, and NPC explained in the economic analysis section (Section IV). The NPV included all the costs for IoT device purchase, drone purchase, initial drone center setup, and yearly operational costs for one drone center. We calculated the cost per hour for purchasing and renting devices using (26) and (27). Initially, the cost per hour for fixed infrastructure was 19 cents. On the other hand, the cost per hour for renting devices was 23 cents. In the next few years, the cost for fixed infrastructure reduced a little. However, it increased since then and surpassed the renting cost after around five years. We saw in the previous results that the cost would be significantly lower if users rent the devices instead of purchasing them. However, one can argue that the user can build his infrastructure, covering the upfront purchasing cost over the years. This analysis shows that the cost per hour would be higher after several years due to the depreciation cost of the devices.

D. Industry Air Pollution Monitoring

We evaluated our proposed model using air pollution monitoring applications where the air quality sensors report data after each specified time interval. For evaluation, we considered that the uniformly distributed sensors have the same distance among them. The time interval of two consecutive readings from each place was 30 min. We conducted our simulation considering the requirement of sensing air quality in 50 and 100 different places in the industrial area. Each drone was equipped with an air quality sensor, and the payload of the sensor data was 315 bytes. We varied the total number of drones and calculated each sensor's energy consumption for reporting the sensed data. We simulated the environment using the same network setup shown in Fig. 10.

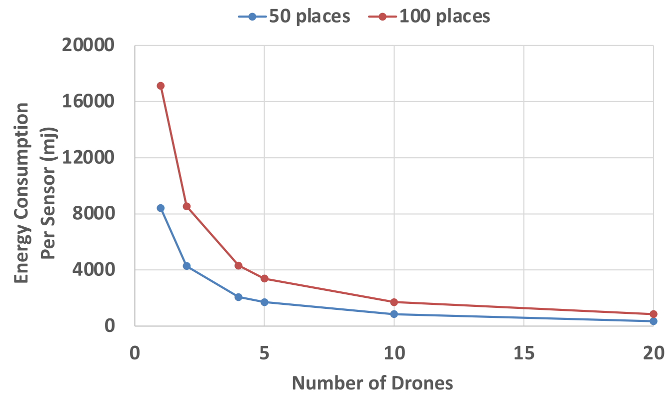

Fig. 16. Energy consumption per sensor with varying number of drones.

Fig. 16 shows the energy cost per sensor considering different numbers of drones. We observed that energy consumption for each sensor decreases with increasing the number of drones. Hence, a lower number of drones can ensure more usage of the sensors. On the other hand, more drones reduce each sensor's energy consumption, which implies less sensor usage. Hence, we can conclude that the drones can sense data from multiple places by reducing the required number of sensors and increase the usage of the sensors.

VI. Related Works and Comparative Discussion

In this section, we compare IoTaaS with the related works regarding drone-based IoT services. Our proposed framework consists of several components: a framework for on-demand provisioning of the IoT devices, optimizing the drone selection, drone payload and coverage area calculation, and economic analysis. All these components are essential for drone-based IoT services for smart cities. To the best of our knowledge, no other research works covered all these issues to provide drone-based IoT services.

Drone-based IoT services gained significant research attention recently. Motlagh *et al.* [14] proposed an UAV-based IoT platform and explained a use case for crowd surveillance. IGaaS by Hoque *et al.* [62] provided an on-demand IoT gateway using drones to improve the quality of service of an IoT network. Vasisht *et al.* [53] proposed a drone-based smart agriculture framework that provides virtual walkthroughs, warnings, and suggestions of different events. Drone-based network packer delivery service in the vehicular network [54], LTE network [63], structural health monitoring [64], edge computing platform for connected autonomous vehicles [65], pipe inspection [66], monitoring hazardous aerial plumes [67], etc., are several other important drone-based IoT applications. Drones can also serve the active IoT devices for collecting data instead of a fixed base station which can reduce the total-transmit power [68]. All these frameworks only focus on collecting data and provide insights for a specific application.

TABLE I
COMPARISON WITH RELATED WORKS

Schemes	Drone-based IoT framework	Drone selection optimization	Economic analysis	Drone payload & Coverage area
Motlagh *et al.* [14]	✓	✗	✗	✗
IGaaS [62]	✓	✗	✓	✗
Vasisht *et al.* [53]	✓	✓	✗	✗
Seliem *et al.* [54]	✓	✗	✗	✓
Skycore [63]	✓	✗	✗	✗
Kalaitzakis *et al.* [64]	✓	✗	✗	✗
Shukla *et al.* [66]	✓	✗	✗	✗
Seiber *et al.* [67]	✓	✗	✗	✗
Mozaffari *et al.* [68]	✓	✓	✗	✗
Zorbas et al. [51]	✓	✓	✗	✗
Bor-Yaliniz *et al.* [69]	✗	✓	✗	✗
Fan et al. [70]	✗	✓	✗	✗
Motlagh *et al.* [52]	✓	✓	✗	✗
Shahzaad *et al.* [71]	✗	✓	✗	✗
Aura [55]	✗	✗	✓	✗
Alwateer *et al.* [56]	✗	✓	✓	✓
Cellcloud [72]	✗	✗	✓	✗
IoTaaS (Proposed Framework)	✓	✓	✓	✓

However, they do not consider relevant issues, such as drone payload, deployment optimization, or billing mechanism.

The optimization of drone selection is an integral part for drone-based IoT applications. Optimization can be performed based on different purposes, such as coverage area, energy consumption, connectivity, task management, etc. Zorbas and Douligeris [51] proposed a mechanism to recharge IoT devices using the battery mounted in drones wirelessly. For this purpose, they have identified a minimum number of drone locations to charge all the nodes. Bor-Yaliniz *et al.* [69] proposed efficient placement of drone-based base stations to optimize the coverage area and revenue. Fan and Ansari [70] proposed a traffic load-balancing scheme using drones to minimize the response time of IoT requests. The research focused on drone placement in suitable areas and allocating traffic load accordingly to avoid traffic congestion. Motlagh *et al.* [52] considered a drone-based IoT platform where they proposed energy and delay aware task assignment mechanism to drones. Different drone-as-a-service mechanisms [71] minimize delivery time and costs based on different heuristics. IoTaaS is complementary to these research works and leverages these different optimization schemes based on the specific IoT applications.

The incentive and billing mechanism is essential for sustainability of a cloud-based service. Aura by Hasan *et al.* [55] is an incentive-driven IoT cloud framework for proximal mobile computation offloading. Alwateer *et al.* [56] proposed drone as a service, where users can rent drones for different purposes. Cellcloud by Al Noor *et al.* [72] is a mobile cloud framework based on bidding incentives where participants can outsource mobile computation resources in exchange for incentives to form a mobile cloud. However, these research works do not complement a billing mechanism for renting IoT devices for sensing purpose.

Overall, the researchers have focused on various drone-based IoT applications for smart cities and relevant optimization schemes. However, little attention has been focused on developing a cloud-based service using drones that enables users to rent IoT devices for on-demand provisioning. A cloud-oriented service for a drone-based IoT platform can resolve the issues by on-demand ad-hoc IoT system deployment while providing the flexibility of cloud services. Table I shows the comparison of our framework with previous works.

VII. CONCLUSION AND FUTURE WORKS

Drones are becoming popular in numerous use cases due to their low price, availability, and usability. They can enable different smart city applications with mounted IoT devices, gateways, and other communication interfaces. This article proposed IoTaaS—a drone-based IoTaaS framework that achieves optimization in terms of deployment, cost, and IoT device usage. The IoT infrastructure service provider can rent out their devices through IoTaaS to achieve maximum resource utilization and ensure the best deals for users to fulfill their requirements. The users can reduce the setup and sensing management issues and costs by renting appropriate IoT devices based on need. Our proof-of-concept implementation showed that IoTaaS could significantly reduce the cost of setting up an IoT system and increase the usage of smart devices. In the future, we plan to enhance the IoTaaS framework to analyze the drone payload and flight time calculation more analytically with different variables, such as various IoT applications, drones, battery capacity, etc. Moreover, we plan to design a broker model for IoTaaS to enable the user to choose the best renting option from multiple service providers.

References

[1] "The Growth in Connected IoT Devices is Expected to Generate 79.4ZB of Data in 2025." International Data Corporation. 2020. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

[2] M. S. Hossain, M. A. Rahman, and G. Muhammad, "Towards energy-aware cloud-oriented cyber-physical therapy system," *Future Gener. Comput. Syst.*, vol. 105, pp. 800–813, Apr. 2020.

[3] M. Alhussein, "Monitoring Parkinson's disease in smart cities," *IEEE Access*, vol. 5, pp. 19835–19841, 2017.

[4] S. Bitam and A. Mellouk, *Bio-Inspired Routing Protocols for Vehicular Ad-Hoc Networks*. London, U.K.: Wiley, 2014.

[5] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, Jan. 2014.

[6] T. Jiang, H. Fang, and H. Wang, "Blockchain-based Internet of Vehicles: Distributed network architecture and performance analysis," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4640–4649, Jun. 2019.

[7] M. S. Mahmud, H. Fang, and H. Wang, "An integrated wearable sensor for unobtrusive continuous measurement of autonomic nervous system," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1104–1113, Feb. 2019.

[8] H. Rajab and T. Cinkelr, "IoT based smart cities," in *Proc. Int. Symp. Netw. Comput. Commun. (ISNCC)*, Rome, Italy, 2018, pp. 1–4.

[9] P. Vlacheas *et al.*, "Enabling smart cities through a cognitive management framework for the Internet of Things," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 102–111, Jun. 2013.

[10] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[11] R. Hasan, M. A. Hoque, Y. Karim, R. Griffin, D. Schwebel, and R. Hasan, "Smartphone-based distracted pedestrian localization using Bluetooth low energy beacons," in *Proc. SoutheastCon*, Raleigh, NC, USA, 2020, pp. 1–2.

[12] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The sensable city: A survey on the deployment and management for smart city monitoring," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1533–1560, 2nd Quart., 2019.

[13] C. Perera, D. S. Talagala, C. H. Liu, and J. C. Estrella, "Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in IoT clouds," *IEEE Trans. Comput. Soc. Syst.*, vol. 2, no. 4, pp. 171–181, Dec. 2015.

[14] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[15] C.-F. Peng, J.-W. Hsieh, S.-W. Leu, and C.-H. Chuang, "Drone-based vacant parking space detection," in *Proc. WAINA*, Krakow, Poland, 2018, pp. 618–622.

[16] P. P. Ray, M. Mukherjee, and L. Shu, "Internet of Things for disaster management: State-of-the-art and prospects," *IEEE Access*, vol. 5, pp. 18818–18835, 2017.

[17] D. Câmara, "Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios," in *Proc. IEEE Conf. Antenna Meas. Appl. (CAMA)*, 2014, pp. 1–4.

[18] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The next big one: Detecting earthquakes and other rare events from community-based sensors," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, Chicago, IL, USA, 2011, pp. 13–24.

[19] E. Mezghani, E. Expósito, and K. Drira, "An autonomic cognitive pattern for smart IoT-based system manageability: Application to comorbidity management," *ACM Trans. Internet Technol.*, vol. 19, no. 1, pp. 1–17, 2019.

[20] D. Analyst. "Why Drones are the Future of the Internet of Things." 2014. [Online]. Available: https://droneanalyst.com/2014/12/01/drones-are-the-future-of-IoT/

[21] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1099–1112, Dec. 2016.

[22] K. Sato and K. Nakashima, "Optimal pricing problem for a pay-per-use system based on the Internet of Things with intertemporal demand," *J. Prod. Econ.*, vol. 221, Mar. 2020, Art. no. 107477.

[23] "An Open Source Operating System for the Internet of Things." Contiki. 2019. [Online]. Available: http://www.contiki-os.org/

[24] T. Bakıcı, E. Almirall, and J. Wareham, "A smart city initiative: The case of barcelona," *J. Knowl. Econ.*, vol. 4, no. 2, pp. 135–148, 2013.

[25] R. Kitchin, "The real-time city? Big data and smart urbanism," *GeoJournal*, vol. 79, no. 1, pp. 1–14, 2014.

[26] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in smart city initiatives: Some stylised facts," *Cities*, vol. 38, pp. 25–36, Jun. 2014.

[27] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proc. ICECC*, Ningbo, China, 2011, pp. 1028–1031.

[28] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[29] W. Li, H. Song, and F. Zeng, "Policy-based secure and trustworthy sensing for Internet of Things in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 716–723, Apr. 2018.

[30] E. Mezghani, E. Exposito, and K. Drira, "A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 3, pp. 224–234, Jun. 2017.

[31] R. Hasan, M. A. Hoque, Y. Karim, R. Griffin, D. Schwebel, and R. Hasan, "StreetBit: A Bluetooth beacon-based personal safety application for distracted pedestrians," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2021, pp. 1–2.

[32] "An IEEE 802.15.4-Based High-Level Communication Protocols for Personal Area Networks." ZigBee. 2016. [Online]. Available: https://zigbee.org/

[33] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals," IETF, RFC 4919, 2007.

[34] M. B. Yassein, W. Mardini, and A. Khalil, "Smart homes automation using Z-wave protocol," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, Agadir, Morocco, 2016, pp. 1–6.

[35] Z.-M. Lin, C.-H. Chang, N.-K. Chou, and Y.-H. Lin, "Bluetooth low energy (BLE) based blood pressure monitoring system," in *Proc. IGBSG*, Taipei, Taiwan, 2014, pp. 1–4.

[36] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Comput. Netw.*, vol. 56, no. 14, pp. 3163–3178, 2012.

[37] J. T. Adams, "An introduction to IEEE STD 802.15.4," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2006, p. 8.

[38] "Amazon Prime Air." Amazon. 2016. [Online]. Available: https://www.amazon.com/b?node=8037720011

[39] A. Varghese, J. Gubbi, H. Sharma, and P. Balamuralidhar, "Power infrastructure monitoring and damage detection using drone captured images," in *Proc. IJCNN*, Anchorage, AK, USA, 2017, pp. 1681–1687.

[40] "Who's Using Amazon Web Services? [2020 Update]." 2020. [Online]. Available: https://www.contino.io/insights/whos-using-aws

[41] H. Liu, Z. Meng, and S. Cui, "A wireless sensor network prototype for environmental monitoring in greenhouses," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.*, Shanghai, China, 2007, pp. 2344–2347.

[42] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Sensor placement with multiple objectives for structural health monitoring," *ACM Trans. Sens. Netw.*, vol. 10, no. 4, pp. 1–45, 2014.

[43] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, 2008.

[44] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment," *IEEE Trans. Comput.*, vol. 56, no. 12, pp. 1681–1695, Dec. 2007.

[45] Y. Xu and X. Yao, "A GA approach to the optimal placement of sensors in wireless sensor networks with obstacles and preferences," in *Proc. CCNC*, vol. 1. Las Vegas, NV, USA, 2006, pp. 127–131.

[46] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and S. Gannouni, "AFIRM: Adaptive forwarding based link recovery for mobility support in NDN/IoT networks," *Future Gener. Comput. Syst.*, vol. 87, pp. 351–363, Oct. 2018.

[47] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 2, pp. 301–311, Feb. 2013.

[48] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 836–848, Jun. 2009.

[49] S. Guo, C. Wang, and Y. Yang, "Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2836–2852, Dec. 2014.

[50] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[51] D. Zorbas and C. Douligeris, "Computing optimal drone positions to wirelessly recharge IoT devices," in *Proc. IEEE INFOCOM WKSHPS*, Honolulu, HI, USA, 2018, pp. 628–633.

[52] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.
[53] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, M. Sudarshan, and S. Stratman, "FarmBeats: An IoT platform for data-driven agriculture," in *Proc. NDSI*, 2017, pp. 515–529.
[54] H. Seliem, R. Shahidi, M. H. Ahmed, and M. S. Shehata, "Drone-based highway-VANET and DAS service," *IEEE Access*, vol. 6, pp. 20125–20137, 2018.
[55] R. Hasan, M. Hossain, and R. Khan, "Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading," *Future Gener. Comput. Syst.*, vol. 86, pp. 821–835, Sep. 2018.
[56] M. Alwateer, S. W. Loke, and N. Fernando, "Drones-as-a-service: A simulation-based analysis for on-drone decision-making," *Pers. Ubiquitous Comput.*, vol. 25, pp. 1–20, Jan. 2021.
[57] F. Metzger, T. Hoßfeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of aggregated IoT traffic and its application to an IoT cloud," *Proc. IEEE*, vol. 107, no. 4, pp. 679–694, Apr. 2019.
[58] E. Walker, "The real cost of a CPU hour," *Computer*, vol. 42, no. 4, pp. 35–41, 2009.
[59] B.-C. Tak, B. Urgaonkar, and A. Sivasubramaniam, "To move or not to move: The economics of cloud computing," in *Proc. HotCloud*, 2011, p. 5.
[60] A. Kamilaris, F. Gao, F. X. Prenafeta-Boldu, and M. I. Ali, "Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications," in *Proc. WF-IoT*, Reston, VA, USA, 2016, pp. 442–447.
[61] "An Introduction to Cooja." Cooja. 2019. [Online]. Available: https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja
[62] M. A. Hoque, M. Hossain, and R. Hasan, "IGaaS: An IoT gateway-as-a-service for on-demand provisioning of IoT gateways," in *Proc. WF-IoT*, New Orleans, LA, USA, 2020, pp. 1–6.
[63] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, "SkyCore: Moving core to the edge for untethered and reliable UAV-based LTE networks," *Commun. ACM*, vol. 64, no. 1, pp. 116–124, 2020.
[64] M. Kalaitzakis, S. R. Kattil, N. Vitzilaios, D. Rizos, and M. Sutton, "Dynamic structural health monitoring using a DIC-enabled drone," in *Proc. ICUAS*, Atlanta, GA, USA, 2019, pp. 321–327.
[65] M. A. Hoque, R. Hasan, and R. Hasan, "R-CAV: On-demand edge computing platform for connected autonomous vehicles," in *Proc. IEEE 7th World Forum Internet Things*, New Orleans, LA, USA, 2021, pp. 65–70.
[66] A. Shukla, H. Xiaoqian, and H. Karki, "Autonomous tracking and navigation controller for an unmanned aerial vehicle based on visual data for inspection of oil and gas pipelines," in *Proc. ICCAS*, Gyeongju, South Korea, 2016, pp. 194–200.
[67] C. Seiber, D. Nowlin, B. Landowski, and M. E. Tolentino, "Tracking hazardous aerial plumes using IoT-enabled drone swarms," in *Proc. IEEE 4th World Forum Internet Things*, Singapore, 2018, pp. 377–382.
[68] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Internet of Things: Can UAVs provide an energy-efficient mobile architecture?" in *Proc. IEEE GLOBECOM*, Washington, DC, USA, 2016, pp. 1–6.
[69] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, 2016, pp. 1–5.
[70] Q. Fan and N. Ansari, "Towards traffic load balancing in drone-assisted communications for IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3633–3640, Apr. 2019.
[71] B. Shahzaad, A. Bouguettaya, S. Mistry, and A. G. Neiat, "Composing drone-as-a-service (DaaS) for delivery," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Milan, Italy, 2019, pp. 28–32.
[72] S. Al Noor, R. Hasan, and M. M. Haque, "CellCloud: A novel cost effective formation of mobile cloud based on bidding incentives," in *Proc. IEEE Cloud*, Anchorage, AK, USA, 2014, pp. 200–207.

Mohammad Aminul Hoque received the B.S. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in March 2016. He is currently pursuing the Ph.D. degree with the Secure and Trust-Worthy Computing Lab, University of Alabama at Birmingham (UAB), Birmingham, AL, USA.

Before joining UAB, he served Therap BD Ltd., Dhaka, as a Software Engineer with the Mobile Development Team from April 2016 to July 2018. His research interests include connected and autonomous vehicles, Internet of Things, and fog/edge computing.

Mahmud Hossain received the Ph.D. degree in computer science from the University of Alabama at Birmingham (UAB), Birmingham, AL, USA, in April 2018.

He is a Senior Cybersecurity Researcher with the Cyber Engineering Department, Visa, Inc., San Francisco, CA, USA. He is also an Adjunct Professor with the Department of Information Science and Technology, George Mason University, Fairfax, VA, USA. Prior to joining Visa, he was a Research Assistant with the Secure and Trustworthy Computing Lab, UAB. He served Samsung R&D Institute Bangladesh, Dhaka, Bangladesh, as a Lead Engineer with the Department of Solution Lab from June 2010 to August 2014. His research interests include security and privacy of Internet of Things, embedded systems, cloud computing, applied cryptography, digital forensics, and blockchain networks.

Shahid Noor received the master's degree in information and communication engineering from the University of Rajshahi, Rajshahi, Bangladesh, in 2007, and the Ph.D. degree in computer science from the University of Alabama at Birmingham, Birmingham, AL, USA, in 2017.

He is an Assistant Professor with the Department of Computer Science, Northern Kentucky University, Highland Heights, KY, USA. He started his career as a Visiting Assistant Professor of Computer Science with St. Olaf College, Northfield, MN, USA, in 2017. His research interests include mobile cloud, Internet of Things, crowdsourcing, and cybersecurity.

S. M. Riazul Islam received the B.Sc. and M.Sc. degrees in applied physics and electronics from the University of Dhaka, Dhaka, Bangladesh, in May 2003 and May 2005, respectively, and the Ph.D. degree in information and communication engineering from Inha University, South Korea, in February 2012.

He is an Assistant Professor with the Department of Computer Science and Engineering, Sejong University, Gwangjin-gu, South Korea. From 2014 to 2017, he worked with the Wireless Communications Research Center, Inha University, Incheon, South Korea, as a Postdoctoral Fellow. From 2016 to 2017, he was also affiliated with Memorial University, St. John's, NL, Canada, as a Postdoctoral Fellow. From 2005 to 2014, he was with the University of Dhaka, Dhaka, Bangladesh, as an Assistant Professor and a Lecturer with the Department of Electrical and Electronic Engineering. In 2014, he worked with the Samsung R&D Institute Bangladesh, Dhaka, as a Chief Engineer with the Department of Solution Lab for advanced research. His research interests include wireless communications, Internet of Things, and applied artificial intelligence.

Ragib Hasan (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in December 2005 and October 2009, respectively.

He is an Associate Professor with the Department of Computer Science, University of Alabama at Birmingham (UAB), Birmingham, AL, USA. Prior to joining UAB, he was an NSF/CRA Computing Innovation Fellow post-doc with the Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA. He is the Founder of the Secure and Trustworthy Computing Lab, UAB.

Dr. Hasan has received multiple awards in his career, including the 2014 NSF CAREER Award, the 2013 Google RISE Award, and the 2009 NSF Computing Innovation Fellowship.