$Psched$: A Priority-Based Service Scheduling Scheme for the Internet of Drones

Cong Pu, *Member, IEEE*, and Logan Carpenter

***Abstract*—With the continuous miniaturization of sensors and processors and ubiquitous wireless connectivity, unmanned aerial vehicles (UAVs), also referred to as drones, are finding many new uses in enhancing our life and paving the way to the realization of Internet of Drones (IoD). In the IoD, a myriad of multisized and heterogeneous drones seamlessly interact with Zone Service Providers (ZSPs) to achieve the goal of assisting drones in accessing controlled airspace and providing navigation services. However, due to the high mobility of drones and the limited communication bandwidth between drones and ZSP, service scheduling becomes a critical issue when a set of drones wants to upload/download data to/from ZSP. In this article, we propose a priority-based service scheduling scheme, also named $Psched$, to provide efficient data upload/download service at ZSP in the IoD. The basic idea is that the $Psched$ objectively and equitably assigns a weight to multiple service scheduling parameters based on multiattribute decision making theory, calculates the serving priority of each service request group, and then serves the service request groups based on the calculated serving priority accordingly. In addition, the $Psched$ takes into account of bandwidth competition between upload and download service requests, and provides a service request balancing to achieve the maximum benefits of service scheduling scheme. In the experimental study, we choose request deadline, data size, and data popularity as service scheduling parameters, and conduct extensive simulation experiments using OMNeT++ for performance evaluation and comparison. The simulation results show that the proposed priority-based service scheduling scheme can not only increase service ratio but also can improve fresh data service ratio and average request serving latency, indicating a viable and efficient approach for satisfying service requests at ZSP in the IoD.**

***Index Terms*—Unmanned aerial vehicles (UAVs), drones, Internet of Drones (IoD), zone service providers (ZSPs), download/upload service scheduling.**

NOMENCLATURE

ω^{ST}	Service cycle.
ω^{RW}	Request reception window.
ω^{UW}	Data upload window.
ω^{DW}	Data download window.
pkt	Service request packet.
n_{id}	Node's identifier.
d_{id}	Identifier of data item.
op	Operation (upload or download) of request.
size	Data size.
dl	Request deadline.
pop	Data popularity.
$\mathrm{RT}^{\mathrm{UL}}$	Upload request table.
$\mathrm{RT}^{\mathrm{DL}}$	Download request table.
$\mathbb{R}^{N\times M}$	Normalized scheduling parameter matrix.
α_j	Coefficient of the jth scheduling parameter.
Entropy	Entropy of the scheduling parameter.
F^{EN}	Entropy proportion.
Γ	Entropy weight of the scheduling parameter.
Pri	Serving priority.
ψ	Share of communication bandwidth for upload request.
$\mathrm{Bef}^{\mathrm{TO}}$	Total benefits gained from serving service requests.
$\mathrm{Bef}^{\mathrm{UL}}$	Benefits of serving upload service requests.
$\mathrm{Bef}^{\mathrm{DL}}$	Benefits of serving download service requests.
$\mathrm{Pen}^{\mathrm{DL}}$	Penalty of serving download request with stale data.
ε	Total number of service request groups in $\mathrm{RT}^{\mathrm{UL}}$.
κ^{ul}	Credits for each served upload service request group.
η	Total number of service request groups in $\mathrm{RT}^{\mathrm{DL}}$.
κ^{dl}	Credits for each served download service request group.
ψ	Optimal value to maximize the total benefits.

Manuscript received March 29, 2020; revised May 22, 2020; accepted May 24, 2020. This work was supported in part by the 2019 Marshall University Summer Undergraduate Creative Discovery and Undergraduate Research Scholar Award, 2020 Marshall University Summer Research Award, and startup grant in the Department of Computer Sciences and Electrical Engineering at Marshall University. Paper no. ISJ-RE-20-09487. *(Corresponding author: Cong Pu.)*

The authors are with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755 USA (e-mail: puc@marshall.edu; carpenter190@marshall.edu).

Digital Object Identifier 10.1109/JSYST.2020.2998010

I. INTRODUCTION

INTERNET of Things (IoT) has received a significant amount of research attention in the last decade, and Edge computing is recently gaining popularity in this context because IoT is becoming common in processing data on the edge of networks [1]. As unmanned aerial vehicles (UAVs), a.k.a. drones, have been increasingly popular among hobbyists, researchers, and investors, there have been a few attempts to integrate drones with the Internet and IoT to build a layered network control architecture designed mainly for assisting drones in accessing controlled airspace and providing navigation services, which is coined with the name Internet of Drones (IoD) [2], [3]. With the on-going miniaturization of sensors and processors, and ubiquitous wireless connectivity and computing infrastructure, drones will find many new uses in a variety of civilian and military application areas [4]. With the help of advanced technologies, we envision that drones in the realm of IoD will totally change the world as we know it.

Inspired by the idea of Air Route Traffic Control Centers created by the Federal Aviation Administration [5], in the architecture of IoD, the airspace is considered as resource that is utilized by drones, and is partitioned into zones that are under authority of one or multiple zone service providers (ZSPs) [6]. In each zone, a number of ZSPs act as routers for drones to access the Internet and provide navigation information in their designated zone to the requesting drones. For example, ZSPs shall provide the drones with the latest weather information[1] such as wind speed and temperature, so drones can successfully take these information into account at the time of planning and executing a trajectory. Since the weather condition frequently changes, it is impossible for drones to preload all the most up-to-date information before traveling. Hence, drones flying into the communication range of ZSP may request the latest information of weather condition locally from ZSP for trajectory guidance.

However, due to the high mobility of drones and the limited communication bandwidth between drones and ZSP, service scheduling becomes a critical issue when a set of drones wants to upload/download data to/from ZSP. Compared to a traditional data access system in which users can always wait for the requested data from the server, drones are moving fast and only stay in the communication range of ZSP for a very short period of time. Thus, there is always a time constraint associated with each data download request from drones. When drones fly out of the communication range of ZSP, the unserved data download requests have to be discarded. Meanwhile, some drones may wish to upload the up-to-date data to ZSP by issuing data upload requests, which compete with data download requests for the limited communication bandwidth. As the number of drones increases, what time for each service request to be served will be critical to the overall performance of service scheduling. Dense placement of ZSPs is a possible solution; however, this approach comes at high deployment and operational costs. Hence, with a limited number of ZSPs, it is important to design an efficient and effective service scheduling scheme for ZSP to satisfy data service requests from drones.

In this article, we design the first solution (to the best of our knowledge) for satisfying data service requests from drones at ZSP in the IoD. Our contribution is summarized in threefold.

- We propose a priority-based service scheduling scheme, named $Psched$. The basic idea is that the $Psched$ objectively and equitably assigns a weight to multiple service scheduling parameters based on multiattribute decision making theory, calculates the serving priority of each service request group, and then serves the service request groups based on the calculated serving priority accordingly. In addition, the $Psched$ takes into account of bandwidth competition between upload and download service requests, and provides a service request balancing to achieve the maximum benefits of the $Psched$.
- We design the $Psched$ with the consideration of multiple service scheduling parameters; thus, the $Psched$ can respond to various service requests with different characteristics nimbly and accurately. In addition, the $Psched$ provides good extensibility and flexibility, and additional service scheduling parameters, such as urgency of service request and fairness of satisfying service request, can be easily added into the $Psched$.
- In the experimental performance evaluation, we choose request deadline, data size, and data popularity as service scheduling parameters, and conduct extensive simulations to validate the effectiveness of the $Psched$. We also revisit prior schemes, such as first come first serve (FCFS), earliest deadline first (EDF), smallest data size first (SDSF), most popular data first (MPDF), and blind service scheduling (BSS), and implement them for performance comparison.

We develop a customized discrete event-driven simulation framework by using OMNeT++ [7] and evaluate its performance through extensive simulation experiments in terms of the download service ratio, upload service ratio, data size service ratio, average request serving latency, fresh data service ratio, energy consumption, bandwidth for upload, and benefits. The simulation results show that the $Psched$ can not only increase service ratio but also can improve fresh data service ratio and average request serving latency, indicating a viable and efficient approach for satisfying data service requests at ZSP.

The rest of the article is organized as follows. Relevant literature are presented and analyzed in Section II. A system model and the proposed service scheduling scheme are provided in Section III. Section IV focuses on simulation results and their analyses. Finally, Section V concludes this article.

II. Related Work

As the emerging of IoD, Flying Ad Hoc Networks (FANETs) are rapidly proliferating and leading to the further development of IoD and its applications. In the past few years, a significant volume of research work has mainly focused on developing routing protocols and communication algorithms for FANETs, such as link-quality and traffic-load aware optimized link state routing protocol [8], aerial network management protocol [9], adaptive hybrid communication protocols [10], jamming-resilient multipath routing protocol [11], data rate and mobility aware routing protocol [12], and their variations. In addition, an overview of the state of the art of routing protocols for FANETs are also presented in [13] and [14]. Through careful analysis and comparison, it is found that each routing protocol has its own definite strengths and weaknesses, and are suitable for a specific situation. Most prior routing approaches focus on the load-carry-and-deliver, the shortest path, the path with the best link quality and lightest traffic load, the geographic position of next-hop node, or mobility prediction. In addition, these routing protocols and communication algorithms assume that a source drone knows the location of destination (drone or ground station) based on the route information, which is accumulated and analyzed by a route discovery or route maintenance algorithm. In our work, instead of addressing the issue of route discovery and maintenance,

[1]In windy conditions, the drone needs much more energy to fly and to compensate for the turbulence. The drones are usually powered by lithium polymer batteries, which drain substantially faster in freezing cold weather.

we emphasize on an efficient data upload/download scheduling algorithm to enhance data accessibility.

Essentially, FANETs are a subcategory of Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs); therefore, the core ideas of data access approaches that were designed for MANETs or VANETs could be used as references. The authors in [15] investigate the usability of providing network connectivity, ultimately, Internet access to mobile users in vehicles. The impact of vehicle's moving speed, average rate of transmission, 802.11 data rate, and size of data packet on throughput, and delay of vehicle-roadside communication are investigated. Unquestionably, the [15] stands out as one of the notable landmarks that sketches a basic picture of how running vehicles contact with RoadSide Unit (RSU) through a "drive-thru" data access. However, it does not discuss the service scheduling feature that provides an efficient way to schedule Internet service to multiple vehicles. In [16], a collection of data files are stored at distributed locations and delivered to passing vehicles. According to the popularity of files, the proposed algorithm schedules the location of files through the selective upload and download of RSUs to maximize the delivery ratio of files to vehicles. However, the extendability of service scheduling model is ignored in [16], where new service scheduling parameters, such as data size, service deadline, and priority, cannot be easily included. In [17], a three-layer vehicular cloud network architecture is designed to support mass data dissemination in real time in VANETs, where two dynamic scheduling algorithms are proposed based on response time and queue length for efficient data scheduling. The proposed dynamic scheduling algorithms are based on predicting the time required for each incoming task, which is very challenging to implement in FANETs.

By employing large-scale channel prediction based on the location information of vehicles, the authors in [18] propose a channel-prediction-based scheduling strategy for cooperative data dissemination in VANETs. In addition, the recursive least-squares algorithm is utilized to achieve the goal of efficient large-scale channel prediction with relatively low computational complexity, which is suitable for real-time processing. Nevertheless, the proposed approach is application specific and employs only a subset of available sensors, mainly the global positioning system (GPS) module or vehicle speed sensors. The authors in [19] propose a service scheduling scheme to transfer safety and nonsafety data in VANETs, where multiple RSUs are connected through wired networks and data are transferred through the collaboration of RSUs. In [20], each vehicle first informs the RSU the list of its current neighboring vehicles and the identifiers of the retrieved and newly requested data. Then, the RSU selects sending and receiving vehicles and corresponding data for vehicle-to-vehicle communication, while it simultaneously broadcasts a data item to vehicles that are instructed to tune into the infrastructure-to-vehicle channel. The authors in [21] propose a novel centralized time-division multiple access (TDMA) based scheduling protocol for practical vehicular networks based on a new weight-factor-based scheduler. In [19]–[21], a push-based communication model based on a centralized scheduler at the RSUs is proposed. However, the pull-based communication, where the provider receives an interest from the consumer before sending contents, is not investigated.

The authors in [22] focus on the scheduling of beaconing periods as an efficient means of energy consumption optimization in FANETs, where a fully distributed learning framework allowing drones to discover their equilibrium beaconing period duration is proposed to achieve the maximum system performance in terms of the encounter rate and energy efficiency. In [23], drones fly over a remote area, where a number of sensors are deployed to collect environmental information, serve as relays for collecting data from remote sensors, thereby improving packet recovery over lossy channels. However, data collection might be frequently interrupted because the drone needs to be recharged. To address this challenge, the base station generates an optimal schedule with guaranteed success rates and balanced energy consumption based on the reception quality reports from drones. In [24], a fleet of drones forms a CDMA-based ad hoc network for tactical surveillance mission. A token circulation scheme is proposed to conduct functions required at the medium access control layer including detection of hidden/lost neighbors, code assignment, and schedule-based cooperative transmission scheduling. In [22]–[24], the scheduling of drone' channel access is studied. However, the involvement of ZSPs in the scheduling decision-making process is not considered.

In [25], a drone scheduling scheme based on a brute-force search combinatorial algorithm is proposed to obtain the optimal scheduling of drones in time to efficiently deploy network service. The scheduling scheme is characterized based on two different states: service execution state and replacement state. In the service execution state, the drones that are equipped with processing and communication devices execute the network functions virtualization to provide the demanded services in the target area. During the replacement phase, the drones with lower battery will be recalled for battery replacement and charging. In [26], a comprehensive survey highlighting drones' potential for the delivery of Internet of Things service from height is provided. In addition, relevant challenges and potential solutions are also described. The authors in [27] survey the work done toward all of the outstanding issues for drone communication networks, such as routing, seamless handover, and energy efficiency.

In summary, various communication and routing algorithms emphasizing on the shortest path, the path with the best link quality and lightest traffic load, the geographic position of next-hop or destination, or the mobility prediction have been well studied in FANETs and similar environments. The reliable communication and routing protocols between drones constitute a building block of the data delivery in each application. To support the growing number of FANET applications and to keep their functioning reliable and stable, the incremental design of routing protocols becomes important. However, the design of routing protocols is beyond the scope of the paper and we leave it for future work. An enormous amount of work has been done to investigate the service scheduling issue at RSUs in VANETs, such as FCFS, longest wait time, most requests first, or longest total stretch first. However, due to the unique characteristics of drones, the service scheduling schemes that were specifically designed for VANETs will fail in the highly dynamic aerial environment. For example, vehicle frequently stops for traffic light at the interaction, where RSU is usually located. Thus, vehicles have a plenty of time to upload/download data to/from RSU through vehicle-to-vehicle and vehicle-to-RSU communications. However, in the IoD, each drone can move very fast in any direction and the communication range of ZSP is limited. As a result, the service request from each drone is only valid for a short time period. If the service request cannot be satisfied before the drone moves out of the communication range of ZSP, it has to be discarded. In the past several years, some work investigated the scheduling issue related to drones. However, their primary concern was how to let drones efficiently access wireless channel and collect data. No efforts (to the best of our knowledge) have been made for designing an efficient solution for data service scheduling at ZSP in the realm of IoD. In addition, those scheduling algorithms only take into account single scheduling parameter, which does not provide good extensibility and flexibility to include additional scheduling parameters.

III. Proposed Priority-Based Service Scheduling Scheme

In this section, we first introduce the system model and assumptions, and then propose a priority-based service scheduling scheme, also named $Psched$, to provide efficient data upload/download service at ZSP in the realm of IoD.

A. System Model and Assumptions

In this article, we consider a number of drones (later nodes) that freely move in an area, where each node is uniquely identified by its node identifier. When the node moves into the communication range of ZSP, it can send upload and/or download service request to ZSP directly. Upload service request indicates that the node is willing to update data stored at ZSP, while download service request is used to retrieve data from ZSP. Here, the ZSP can act as either a buffer point between nodes or a gateway to the Internet. In order to extend the operational lifetime, each node is equipped with rechargeable batteries to achieve energy self-sufficiency through recharging from wireless recharging stations [28] or environmental energy resources [29]. Since each node can move very fast in any direction and the communication range of ZSP is limited, the service request from each node is only active for a short period of time. When the node moves out of the communication range of ZSP, the service request that has not been satisfied or served must be discarded. We also assume that each node is equipped with the GPS, inertial navigation system (INS), and inertial measurement units (IMUs) to obtain its current geographical position and mobility information [30]. Thus, when the node enters the communication range of ZSP, it can estimate the time when it leaves the communication range of ZSP, which is considered as the service request deadline.

B. Brief Overview

A set of nodes that are in the communication range of ZSP can update (or retrieve) data at (or from) ZSP through the upload (or download) service request. The nodes that do not have direct communication with ZSP can employ multihop relay protocols [13], [14] to update/retrieve data at/from ZSP; however, this is not within the scope of this article. The ZSP maintains two simple data structures, upload request queue and download request queue, to temporarily store all received upload and download service requests, respectively. Before running the scheduling algorithm, the ZSP examines upload and download request queues, respectively, and groups the received service requests that are interested in the same data item. Then, the ZSP objectively and equitably assigns a weight to multiple service scheduling parameters based on multiattribute decision making theory, calculates the serving priority of each service request group, and then serves the service request groups based on the calculated serving priority accordingly.

In addition, upload service requests compete the limited communication bandwidth with other download service requests, some download service requests may get stale data if certain upload service requests are not served, degrading the overall quality of scheduling service. In light of these, the ZSP takes into account of bandwidth competition between upload and download service requests, and provides a service request balancing to achieve the maximum benefit of service scheduling scheme. More details about the proposed priority-based service scheduling scheme are presented below. Nomenclature section lists all notations used in this article.

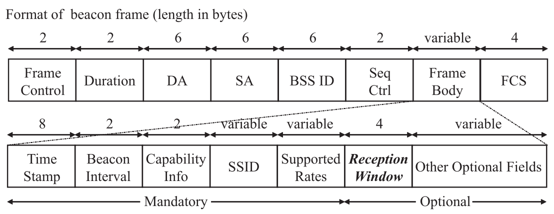

Fig. 1. Revised format of IEEE 802.11 beacon frame, where the field of *Reception Window* indicates when the request reception window ends.

C. Service Request Scheduling

First, the ZSP maintains a service cycle ω^{ST}, which consists of three consecutive windows: request reception window ω^{RW}, data upload window ω^{UW}, and data download window ω^{DW}. Within request reception window ω^{RW}, the ZSP periodically broadcasts beacon packets to advertise when the request reception window will end. The format of beacon packet is shown in Fig. 1. Here, ω^{RW} is a system parameter and can be adaptively set according to the density of nodes in the surrounding area. For example, a larger size of ω^{RW} can be given when the density of nodes is high in the local area, so more nodes can send upload/download service requests to ZSP. ω^{UW} and ω^{DW} are dynamically adjusted in every service cycle based on the number of received upload and download service requests and historical information, more details are presented in Section III-D. When the node enters the communication range of ZSP, it first listens to the wireless medium for a beacon packet. If the request reception window is still open, the node can send upload/download service request to ZSP. Otherwise, it has to wait until the next service cycle.

TABLE I
UPLOAD AND DOWNLOAD SERVICE REQUEST TABLE TEMPLATE

Data Identifier	Data Size$^{\aleph}$	Group Request Deadline$^{\Im}$	Data Popularity$^{\wp}$
d_1	X_{11}	X_{12}	X_{13}
d_2	X_{21}	X_{22}	X_{23}
d_3	X_{31}	X_{32}	X_{33}
...	...	...	...
d_N	X_{N1}	X_{N2}	X_{N3}

$^{\aleph}$: X_{i1} is the size size$_i$ of data item d_i, i ∈ [1, N].
$^{\Im}$: X_{j2} is the group request deadline dl$_j$ of data item d_j, j ∈ [1, N].
$^{\wp}$: X_{k3} is the data popularity pop$_k$ of data item d_k, k ∈ [1, N].

Second, each service request packet pkt $[n_{id}, d_{id}, \text{op}, \text{dl}]$ contains node's identifier (n_{id}), the identifier of data item (d_{id}), the operation (upload or download) of request (op), and the request deadline (dl), respectively. Here, the request deadline dl is the critical time constraint of service request and indicates that the request must be satisfied or served before the deadline expires. When the ZSP receives upload and download service requests, it first puts upload and download service requests into the upload request queue and download request queue, respectively. When the request reception window ends, the ZSP examines upload and download request queues, groups the service requests that are interested in the same data item, and then builds upload request table and download request table, respectively. Here, the template of upload request table RT$^{\text{UL}}$ and download request table RT$^{\text{DL}}$ is listed in Table I, where each entry consists of four components: the identifier of data item, the size of data item, the group request deadline, and the data popularity. The group request deadline is the earliest deadline in the request group, where all requests are interested in the same data item, and the number of requests in the same request group is the measurement of data popularity.

Third, the ZSP retrieves the information of the upload request table RT$^{\text{UL}}$ and download request table RT$^{\text{DL}}$, removes the column of data identifier, normalizes the value of each scheduling parameter (data size, group request deadline, and data popularity) in RT$^{\text{UL}}$ and RT$^{\text{DL}}$ based on the following:

$$X^*_{ij} = \frac{\max\{X_j\} - X_{ij}}{\max\{X_j\} - \min\{X_j\}} \times \alpha_j + (1 - \alpha_j)$$
$$i \in [1, N] \quad j \in [1, 3] \tag{1}$$

and then generates normalized upload and download scheduling parameter matrix, respectively. The format of the normalized scheduling parameter matrix $\mathbb{R}^{N\times 3}$ is shown as

$$\mathbb{R}^{N\times 3} = \begin{bmatrix} X^*_{11} & X^*_{12} & X^*_{13} \\ X^*_{21} & X^*_{22} & X^*_{23} \\ X^*_{31} & X^*_{32} & X^*_{33} \\ \cdots & \cdots & \cdots \\ X^*_{N1} & X^*_{N2} & X^*_{N3} \end{bmatrix}$$

where *N* and 3 indicate the number of entries in the normalized scheduling parameter matrix and three scheduling parameters (data size, group request deadline, and data popularity), respectively. Here, α_j is the coefficient used to control the value range and adjust the effect of the jth scheduling parameter for subjective preference, and $\sum_{j=1}^{3} \alpha_j = 1.0$. The rationale behind this operation is to reduce any negative effect from different metrologies, since data size, group request deadline, and data popularity have different measurement scales and units. After that, the ZSP calculates the entropy of the jth scheduling parameter according to the following:

$$\text{Entropy}_j = -\frac{1}{\ln N} \sum_{i=1}^{N} \left(F^{EN}_{ij} \cdot \ln F^{EN}_{ij} \right), \quad j \in [1, 3] \tag{2}$$

where F^{EN}_{ij} is the entropy proportion of the jth scheduling parameter associated with service request group d_i, and is represented as

$$F^{EN}_{ij} = \frac{X^*_{ij}}{\sum_{k=1}^{N} X^*_{kj}}, \quad j \in [1, 3]. \tag{3}$$

Based on the theory concept of multiattribute decision making [31] and entropy [32], the entropy weight of the jth scheduling parameter Γ_j can be defined as

$$\Gamma_j = \frac{1 - \text{Entropy}_j}{\sum_{k=1}^{3} (1 - \text{Entropy}_k)}, \quad j \in [1, 3]. \tag{4}$$

Finally, the serving priority of the service request group d_i, denoted by Pri$_i$, can be obtained from

$$\text{Pri}_i = 1 - \frac{\sum_{j=1}^{3} (\Gamma_j \cdot X^*_{ij})}{\sum_{k=1}^{N} \sum_{j=1}^{3} (\Gamma_j \cdot X^*_{kj})}, \quad i \in [1, N]. \tag{5}$$

Please note, the service request group is used interchangeably with the data item in the upload and download request table.

Finally, the ZSP ranks the calculated serving priority of service request groups in the upload request table RT$^{\text{UL}}$ and download request table RT$^{\text{DL}}$, and then serves each service request group in RT$^{\text{UL}}$ and RT$^{\text{DL}}$, respectively. In this article, the data upload window is scheduled in front of the data download window; thus, the upload service requests will be served first. The rationale behind this design is that some upload service requests may update the data items that are currently being requested or will be requested by other nodes. By serving the upload service requests earlier, the data item will be updated first, and the download service requests can retrieve the up-to-date data items. In addition, the same service scheduling scheme is used for both upload and download service requests. However, our approach is designed with the consideration of extensibility and flexibility for potential extension and improvement in the future, since an individual service request table is used for upload and download service requests, respectively. The benefit of dual tables is that upload and download requests can have their own scheduling scheme, and we only need to compare the upload service request table and download service request table instead of individual upload and download requests. Intuitively, if two or more service request groups have the same serving priority, the following rules will be applied.

- First, the deadline is utilized to decide the serving order, and the one with an earlier deadline will be served first.

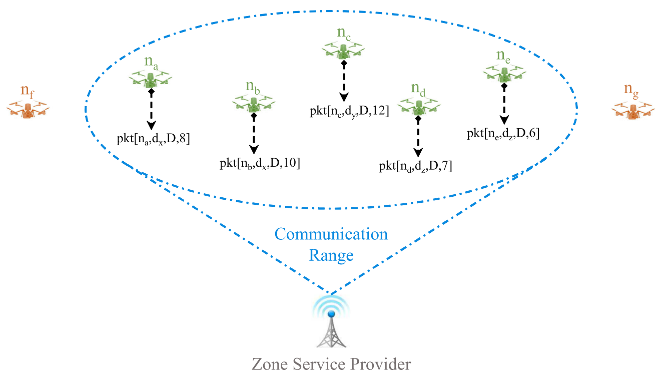

Fig. 2. Snapshot of service scheduling at ZSP, where green nodes are in the communication range of ZSP while orange nodes are not. Downward arrow represents the service request packet and *D* in the pkt indicates the download service request.

TABLE II
DOWNLOAD SERVICE REQUEST TABLE

Data Identifier	Data Size	Group Request Deadline	Data Popularity
d_x	3.5 MB	8 sec	2
d_y	1.1 MB	12 sec	1
d_z	2.2 MB	6 sec	2

- Second, the data popularity is utilized to decide the serving order, and the one with a larger data popularity will be served first.
- Third, the data size is utilized to decide the serving order, and the one with a smaller size data will be served first.

For example, as shown in Fig. 2, node n_a, n_b, n_c, n_d, and n_e send download service request packets to ZSP to ask for data item d_x, d_y, and d_z, respectively, where the data size of d_x, d_y, and d_z is assumed to be 3.5, 1.1, and 2.2 MB. Thus, the ZSP builds download service request table, as listed in Table II, and generates the normalized scheduling parameter matrix $\mathbb{R}^{3\times3}$, as shown below, according to (1). In this example, the value of a_1, a_2, and a_3 is set to 0.30, 0.35, and 0.35, respectively.

$$\mathbb{R}^{3\times3} = \begin{bmatrix} 0.7 & 0.8833 & 0.65 \\ 1.0 & 0.65 & 1.0 \\ 0.8626 & 1.0 & 0.65 \end{bmatrix}$$

After that, the ZSP calculates the entropy proportion and the entropy weight of the data size, group request deadline, and data popularity according to (2) and (4), such as $\text{Entropy}_{\text{size}} = 0.9905$, $\text{Entropy}_{\text{dead}} = 0.9860$, $\text{Entropy}_{\text{pop}} = 0.9796$, and $\Gamma_{\text{size}} = 0.2164$, $\Gamma_{\text{dead}} = 0.3189$, $\Gamma_{\text{pop}} = 0.4647$. Then, the ZSP calculates the serving priority of each service request group in Table II based on (5), $\text{Pri}_x = 0.6975$, $\text{Pri}_y = 0.6345$, and $\text{Pri}_z = 0.6677$. Finally, the ZSP ranks the calculated serving priority of each service request group d_x, d_y, and d_z, 0.6975, 0.6345, and 0.6677, and then broadcasts d_x first, d_z second, and d_y third.

D. Service Request Balancing

Since upload service requests and download service requests compete for the limited communication bandwidth, if the ZSP fails to serve certain upload service requests, some data may not be updated timely and download service requests may get stale data, which degrades the overall quality of data service. Thus, it is important to design a service request balancing scheme to achieve the maximum benefit of service scheduling through balancing the served upload and download service requests. Suppose that ψ is the allocated share of communication bandwidth for upload service requests, and the rest $1 - \psi$ is the share for download service requests. Let Bef^{TO} be the total benefits gained from the served upload and download service requests, which is the sum of the benefits of served upload service requests Bef^{UL}, the benefits of served download service requests Bef^{DL}, and the penalty of served download service request group with stale data Pen^{DL}. Then Bef^{To} is expressed as

$$\text{Bef}^{\text{TO}} = \text{Bef}^{\text{UL}} + \text{Bef}^{\text{DL}} + \text{Pen}^{\text{DL}}. \tag{6}$$

First, Bef^{UL} can be expressed as

$$\text{Bef}^{\text{UL}} = \varepsilon \cdot \psi \cdot \kappa^{\text{ul}} \tag{7}$$

where ε is the total number of service request groups in the upload request table RT^{UL} and κ^{ul} is the credits for each served upload service request group. Second, Bef^{DL} is expressed as

$$\text{Bef}^{\text{DL}} = \eta \cdot (1-\psi) \cdot \psi \cdot \kappa^{\text{dl}} \tag{8}$$

where η is the total number of service request groups in the download request table RT^{DL} and κ^{dl} are the credits for each served download service request group. Third, Pen^{DL} is expressed as

$$\text{Pen}^{\text{DL}} = \eta \cdot (1-\psi) \cdot (1-\psi) \cdot (-\kappa^{\text{dl}}) \tag{9}$$

where $-\kappa^{\text{dl}}$ is the penalty of each served download service request group with stale data. Finally, the total benefits Bef^{TO} can be represented as

$$\begin{aligned}\text{Bef}^{\text{TO}} &= \varepsilon \cdot \psi \cdot \kappa^{\text{ul}} + \eta \cdot (1-\psi) \cdot \psi \cdot \kappa^{\text{dl}} + \eta \cdot (1-\psi)^2 \\ &\quad \cdot (-\kappa^{\text{dl}}) = -2 \cdot \eta \cdot \kappa^{\text{dl}} \cdot \psi^2 + (\varepsilon \cdot \kappa^{\text{ul}} + 3 \cdot \eta \cdot \kappa^{\text{dl}}) \\ &\quad \cdot \psi - \eta \cdot \kappa^{\text{dl}}.\end{aligned} \tag{10}$$

We can calculate the optimal value of ψ to maximize the total benefits by solving the quadratic function with the linear constraint on ψ. The optimal solution is

$$\psi = \begin{cases} \min(\frac{\varepsilon \cdot \kappa^{\text{ul}} + 3 \cdot \eta \cdot \kappa^{\text{dl}}}{4 \cdot \eta \cdot \kappa^{\text{dl}}}, 1), & \text{if } (\varepsilon > 0) \wedge (\eta > 0) \\ 0, & \text{if } (\varepsilon = 0) \wedge (\eta > 0) \\ 1, & \text{if } (\varepsilon > 0) \wedge (\eta = 0). \end{cases} \tag{11}$$

When $\varepsilon = 0$, which means the ZSP does not receive any upload service request, the communication bandwidth is totally allocated to download service requests. On the other side, when $\eta = 0$, which means the ZSP does not receive any download service request, the communication bandwidth is totally allocated to upload service requests. When $\varepsilon > 0$ and $\eta > 0$, the communication bandwidth allocation totally depends on the number of received upload and download service request groups.

According to (11), the optimal value of ψ depends on the number of upload and download service request groups, which are related to the service request workload. However, the service

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Network area	1000×1000 m^2
Number of nodes	50 to 200
Communication range of node	250 meters
Communication range of ZSP	200 meters
Moving speed	25 meter/sec
Mobility model	Random waypoint
Upload/download service request rate	0.125 packet/sec
The size of data set	50
Date size	50 KB to 5 MB
Simulation time	10000 seconds

request workload changes all the time, the value of ψ should be dynamically and adaptively adjusted according to the service request workload in each service cycle and the historical information. Thus, the optimal share of communication bandwidth for upload service request groups in the kth service cycle, denoted as ψ_k, is updated by the low-pass filter with a filter gain constant σ and represented as

$$\psi_k = \sigma \cdot \psi_{k-1} + (1-\sigma) \cdot \psi_{\text{avg}}. \quad (12)$$

ψ_{k-1} is the measurement in the $k-1$th service cycle, and ψ_{avg} is the average value of optimal share of all past service cycles and calculated as $\frac{\sum_{i=1}^{k-1}\psi_i}{k-1}$. Based on the calculated ψ_k, the ZSP chooses and serves the corresponding number of upload and download service request groups, respectively. Major operations of the $Psched$ are summarized in Fig. 3.

```
Notations:
• $RT^{UL}$, $RT^{DL}$, $pop$, $n_{id}$, $d_{id}$, $op$, $dl$, $\omega^{ST}$, $\omega^{RW}$, $\omega^{UW}$, $\omega^{DW}$, $a_j$,
  $\mathbb{R}^{N\times M}$, $Entropy_j$, $\Gamma_j$, $Pri_i$, and $\psi$: Defined before.
• $RT^{UL}[j].metrics$: The information of scheduling parameter is associated
  with service request group $d_j$ in the upload request table. Here, metrics is
  data size ($size$), request deadline (dl), or data popularity (pop).
• $pkt[n_{id}, d_{id}, op, dl]$: A data service request packet containing a node
  identifier ($n_{id}$), the identifier of data item ($d_{id}$), the operation (upload or
  download) of request (op), and the request deadline (dl). Here, op is either
  upload, denoted as U, or download, denoted as D.
Service Request Scheduling at ZSP:
◇ When ZSP receives service request packet $pkt[n_i, d_x, op, dl]$ from $n_i$:
  if op is U /∗ Receive upload service request ∗/
    if $d_x \in RT^{UL}$ /∗ $d_x$ has been requested before ∗/
      $RT^{UL}[x].pop$ += 1; /∗ Increase data popularity ∗/
      if $pkt[dl] < RT^{UL}[x].dl$ /∗ pkt has earlier deadline ∗/
        $RT^{UL}[x].dl = pkt[dl]$; /∗ Update request group deadline ∗/
    else /∗ $d_x$ is a newly requested data ∗/
      $RT^{UL} = RT^{UL} \cup [d_x, size, dl, pop]$;
  else /∗ Receive download service request ∗/
    if $d_x \in RT^{DL}$ /∗ $d_x$ has been requested before ∗/
      $RT^{DL}[x].pop$ += 1; /∗ Increase data popularity ∗/
      if $pkt[dl] < RT^{DL}[x].dl$ /∗ pkt has earlier deadline ∗/
        $RT^{DL}[x].dl = pkt[dl]$; /∗ Update request group deadline ∗/
    else /∗ $d_x$ is a newly requested data ∗/
      $RT^{DL} = RT^{DL} \cup [d_x, size, dl, pop]$;
◇ When the request reception window $\omega^{RW}$ ends:
  for i ∈ M /∗ each scheduling parameter in $RT^{UL}$ and $RT^{DL}$ ∗/
    for j ∈ N /∗ each entry in $RT^{UL}$ and $RT^{DL}$ ∗/
      $X^*_{ij} = \frac{max\{X_j\}-X_{ij}}{max\{X_j\}-min\{X_j\}} \times \alpha_j + (1-\alpha_j)$;
  for i ∈ N /∗ each entry in $RT^{UL}$ and $RT^{DL}$ ∗/
    for j ∈ M /∗ each scheduling parameter in $RT^{UL}$ and $RT^{DL}$ ∗/
      $Entropy_j = -\frac{1}{\ln N}\sum_{i=1}^{N}\left(F^{EN}_{ij} \cdot \ln F^{EN}_{ij}\right)$;
      $\Gamma_j = \frac{1-Entropy_j}{\sum_{k=1}^{3}(1-Entropy_k)}$;
    $Pri_i = 1 - \frac{\sum_{j=1}^{3}(\Gamma_j \cdot X^*_{ij})}{\sum_{k=1}^{N}\sum_{j=1}^{3}(\Gamma_j \cdot X^*_{kj})}$;
  Sort($Pri_i$, $RT^{UL}$); Sort($Pri_i$, $RT^{DL}$);
Service Request Balancing at ZSP:
  if $(\varepsilon > 0) \wedge (\eta > 0)$
    $\psi = min(\frac{\varepsilon \cdot \kappa^{ul} + 3 \cdot \eta \cdot \kappa^{dl}}{4 \cdot \eta \cdot \kappa^{dl}}, 1)$;
  if $(\varepsilon = 0) \wedge (\eta > 0)$
    $\psi = 0$;
  if $(\varepsilon > 0) \wedge (\eta = 0)$
    $\psi = 1$;
ZSP Serves Upload and Download Service Requests:
◇ When the data upload window $\omega^{UW}$ begins:
  Serve($\psi$, $RT^{UL}$);
◇ When the data download window $\omega^{DW}$ begins:
  Serve($1 - \psi$, $RT^{DL}$);
```

Fig. 3. Pseudocode of the proposed $Psched$.

IV. PERFORMANCE EVALUATION

A. Simulation Testbed

We develop an OMNeT++ [7] based simulation framework to conduct simulation experiments and evaluate the proposed priority-based service scheduling scheme. The experimental simulation is conducted in a 1000 × 1000 (m^2) square network area, where 50–200 nodes are uniformly and randomly distributed at the beginning of simulation. Each node is assumed to have the same constant and low altitude during the flight, i.e., 150 m. An IEEE 802.11p radio transceiver is equipped by each node and the transmission range is 250 m. The communication area of ZSP is set to 200 m. Before entering the communication range of ZSP, the movement of each node is based on the random waypoint mobility model [33], where each node moves toward a randomly selected destination in the square network area with a constant speed of 25 m/s and a zero pause time. After moving into the communication range of ZSP, the node maintains the direction of flight with the same speed until it leaves the communication range of ZSP. Each node generates the upload or/and download service requests at the rate of 0.125 packet/s. The size of a dataset is 50 and the size of each data item is from 50 kB to 5 MB. The total simulation time is set to 10 000 s, and each simulation scenario is repeated ten times with different randomly generated seeds to obtain steady performance results. The simulation parameters are summarized in Table III. In this article, we measure the performance of the proposed scheduling scheme in terms of a download service ratio, upload service ratio, data size service ratio, average request serving latency, fresh data service ratio, energy consumption, bandwidth for upload, and benefits. We also revisit and implement prior schemes, FCFS, EDF, SDSF, MPDF, and BSS for performance comparison.

1) FCFS: The service request with the earliest arrival time will be served first.
2) EDF: The service request with the shortest deadline will be served first.
3) SDSF: The service request with the smallest requested data size will be served first.
4) MPDF: The service request asking for the most popular data will be served first.
5) BSS: The service request will be served randomly.

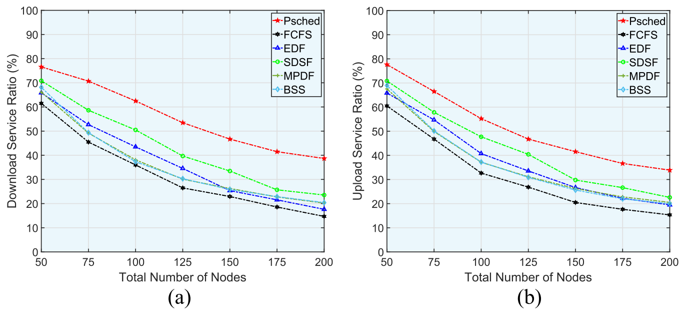

Fig. 4. Performance of download service ratio and upload service ratio against total number of nodes.

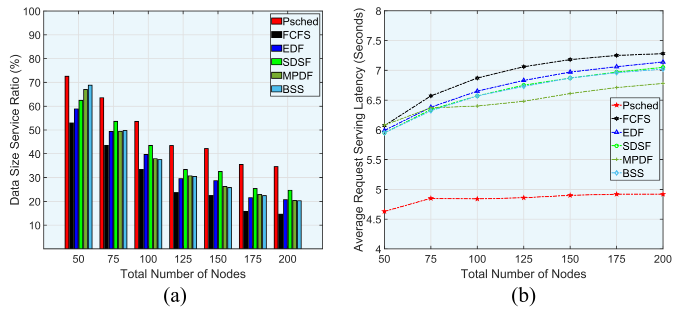

Fig. 5. Performance of data size service ratio and average request serving latency against total number of nodes.

B. Simulation Results and Analysis

In Fig. 4, the download service ratio and upload service ratio are measured by varying the total number of nodes. As shown in Fig. 4(a), the download service ratio of $Psched$ is higher than that of FCFS, EDF, SDSF, MPDF, and BSS. Since the $Psched$ groups the download service requests that ask for the same data item into one service request group, more download service requests can be served simultaneously with one broadcast of data item, resulting in the increment of download service ratio. However, the FCFS, EDF, SDSF, MPDF, and BSS do not group the download service requests. For multiple service requests asking for the same data item, the ZSP has to broadcast the data item multiple times, which consumes the communication bandwidth. As a result, a lower download service ratio is observed. As the total number of nodes increases, the performance of the download service ratio of six schemes decrease linearly. This is because more nodes will send download service requests to ZSP at the same time, which are competing for the limited communication bandwidth. As a result, less number of download service requests can be served by ZSP, and the download service ratio decreases. In Fig. 4(b), the upload service ratio of $Psched$ decreases as the total number of nodes increases. However, the $Psched$ still outperforms FCFS, EDF, SDSF, MPDF, and BSS. Since the $Psched$ can dynamically balance upload and download service requests, more upload service requests can be served compared to that of FCFS, EDF, SDSF, MPDF, and BSS, which leads to a higher upload service ratio. The FCFS, EDF, SDSF, MPDF, and BSS do not distinguish upload and download service requests. After ranking or randomizing the received upload and download service requests, the ZSP serves the service requests according to the ranking. However, if the ZSP receives more download service requests, more communication bandwidth will be used to serve the download service requests. As a result, the upload service ratio is decreasing.

In Fig. 5, the data size service ratio and average request serving latency are observed with varying total number of nodes. Here, the data size service ratio is calculated as the total size of served data items divided by the total size of requested data items, and the average request serving latency is calculated as the total upload and download requests serving latency divided by the total number of upload and download service requests. As shown in Fig. 5(a), the performance of $Psched$ is significantly better than FCFS, EDF, SDSF, MPDF, and BSS. Since the $Psched$ takes into account of data size as one of scheduling metrics to serve service requests, the service requests who are interested in larger data items have more chances to be served first. Thus, the data size service ratio significantly increases. However, the FCFS, EDF, MPDF, and BSS do not consider the data size for serving service requests, and the SDSF serves the service request that asks for the smallest data item. As the total number of nodes increases, the overall data size service ratio decreases because of service requests competing for the limited communication bandwidth. In Fig. 5(b), as the total number of nodes increases, the average request serving latency of FCFS, EDF, SDSF, MPDF, BSS, and $Psched$ increase. This is because more nodes can send more upload and download service requests, and more service requests cannot get satisfied and experience the longest latency due to the limited communication bandwidth and service request deadline. However, the average request serving latency of $Psched$ is much lower than that of FCFS, EDF, SDSF, MPDF, and BSS. In the $Psched$, the service request deadline is considered as one of the service scheduling parameters. This can guarantee that most service requests can be satisfied before the deadline, which results in a lower average request serving latency.

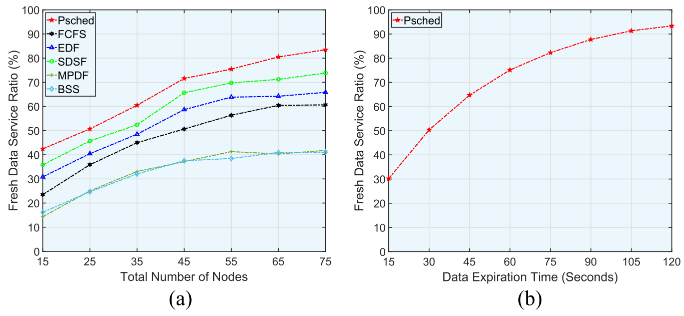

Fig. 6. Performance of fresh data service ratio against total number of nodes and data expiration time.

In Fig. 6, we measure the fresh data service ratio by changing the total number of nodes and data expiration time. As shown in Fig. 6(a), the fresh data service ratio of $Psched$ is higher than that of FCFS, EDF, SDSF, MPDF, and BSS because the $Psched$ considers the service request deadline for scheduling. Thus, more download service requests can be served with fresh data items before the deadline expires, and a higher fresh data service ratio is observed. In Fig. 6(b), when the data expiration time is

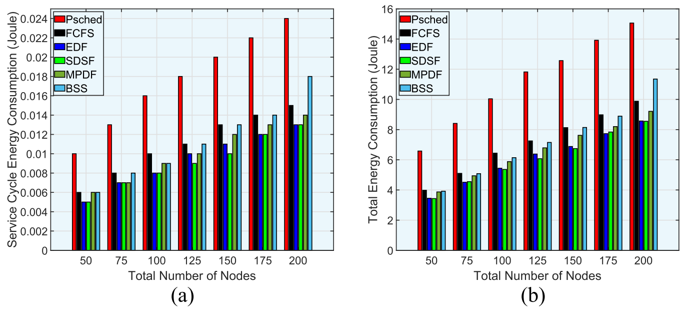

Fig. 7. Performance of energy consumption against total number of nodes.

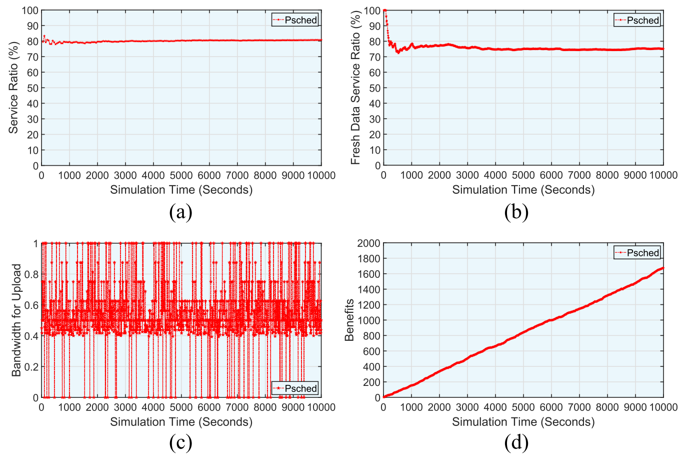

Fig. 8. Performance of service ratio, fresh data service ratio, bandwidth for upload, and benefits against simulation time.

extended, the fresh data service ratio of *Psched* is significantly increased. This is because the data items will be valid for a longer time period with a larger expiration time, the *Psched* can serve more download service requests with valid data items, resulting in a significant increase in the fresh data service ratio of *Psched*.

In Fig. 7, the service cycle energy consumption and total energy consumption are observed by varying the total number of nodes. Here, the energy consumption of scheduling algorithms are measured based on energy consumption per CPU cycle [34]. Overall, the service cycle and total energy consumption of *Psched* are higher than that of FCFS, EDF, SDSF, MPDF, and BSS. The FCFS, EDF, SDSF, MPDF, and BSS simply rank or randomize the upload and download service requests to decide the order of serving service requests. But, the *Psched* is designed based on multiattribute decision-making theory, which is much more complex than FCFS, EDF, SDSF, MPDF, and BSS. As a result, more energy consumption is observed by *Psched*. However, in this article, we assume that the ZSP has sustainable and infinite energy supply. Thus, the energy consumption of the ZSP is not a major concern of the article.

In Fig. 8, we measure the performance of the service ratio, fresh data service ratio, bandwidth for upload, and benefits against simulation time. In Fig. 8(a), the overall service ratio of *Psched* can be maintained around 80% over the entire simulation period. Since the *Psched* considers multiple service scheduling parameters and serves the service requests based on the serving priority accordingly, more service requests can be served and a higher service ratio is observed. In Fig. 8(b), the fresh data service ratio can be observed around 75% when the total number of nodes is set to 75. However, we expect that the fresh data service ratio will be lower if the total number of nodes increases. This is because more nodes will send more download service requests to compete for the limited communication bandwidth, some download service requests might get stale data items or even would not be able to get served. In Fig. 8(c), the bandwidth for upload service requests is observed as the simulation time elapses. Overall, the bandwidth for upload service requests is fluctuating around 0.5, which indicates that the upload service requests and download service requests cooperatively use the limited communication bandwidth. However, at certain time, the number of upload service requests is too small, the entire bandwidth will be utilized by the download service requests, vice versa. In Fig. 8(d), the benefits of upload service requests and download service requests are measured against the simulation time. As the simulation time elapses, the benefits of service request balancing is increasing linearly, indicating that the *Psched* continuously and positively balances the limited communication bandwidth for upload service requests and download service requests.

V. Conclusion

In this article, we investigated the service scheduling challenges between drones and ZSP in the IoD. We proposed a priority-based service scheduling scheme called *Psched* to make data upload/download service scheduling decision with the consideration of data request deadline, data size, and data popularity. In order to address the issue of bandwidth competition between upload and download service requests, we also proposed a service request balancing scheme to achieve the maximum benefits of service scheduling. Extensive simulation results show that the proposed priority-based service scheduling scheme can improve the service ratio, data size service ratio, average request serving latency as well as fresh data service ratio, indicating a viable and efficient approach for satisfying service requests in the IoD.

As a future work, we plan to extend the *Psched* by including a multihop relay technique, so that the drones that are not in the communication range of ZSP can request data service as well. The primary goal is to design a lightweight packet forwarding algorithm based on the combination of multiple real-time network metrics such as link quality, hop count, and residual energy to efficiently forward service request packets in the aerial environment. In addition, we also plan to investigate the issue of GPS inaccuracy related to drone mobility and propose a trajectory prediction scheme. The basic idea of the trajectory prediction scheme is to predict the trajectory of a drone until a future point in time. The predicted position at the prediction time is calculated by linear extrapolation of the current geographic position and mobility information of the drone. Finally, since radio propagation and its channel dynamics cannot easily be captured by simulation models, we plan to develop a small-scale testbed with small and safe quad-copters, e.g., Crazyflie 2.0, and deploy a real outdoor environment to see the full potential of the proposed scheme.

Acknowledgment

The authors would like to thank the anonymous reviewers of IEEE Systems Journal for providing their constructive comments to improve the overall quality of this article.

References

[1] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 110–115, Aug. 2018.

[2] W. Ji, J. Xu, H. Qiao, M. Zhou, and B. Liang, "Visual IoT: Enabling Internet of Things visualization in smart cities," *IEEE Net.*, vol. 33, no. 2, pp. 102–110, Mar. 2019.

[3] A. Koubâa *et al.*, "Dronemap planner: A service-oriented cloud-based management system for the Internet-of-Drones," *Elsevier Ad Hoc Netw.*, vol. 86, pp. 46–62, Apr. 2019.

[4] A. Savkin and H. Huang, "A method for optimized deployment of a network of surveillance aerial drones," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4474–4477, Dec. 2019.

[5] [Online]. Available: https://www.cfinotebook.net/notebook/air-traffic-control/air-route-traffic-control-center, Accessed on: May 22, 2020.

[6] S. Jangirala, A. Das, N. Kumar, and J. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[7] A. Varga, *OMNeT++*. 2014. [Online]. Available: http://www.omnetpp.org/

[8] C. Pu, "Link-quality and traffic-load aware routing for UAV ad hoc networks," in *Proc. IEEE CIC*, Oct. 2018, pp. 71–79.

[9] G. Secinti, P. Darian, B. Canberk, and K. Chowdhury, "SDNs in the sky: Robust end-to-end connectivity for aerial vehicular networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 16–21, Jan. 2018.

[10] Z. Zheng, A. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 136–142, Jan. 2018.

[11] C. Pu, "Jamming-resilient multipath routing protocol for flying ad hoc networks," *IEEE Access*, vol. 6, pp. 68 472–68 486, Nov. 2018.

[12] K. Darabkh, M. Alfawares, and S. Althunibat, "MDRMA: Multi-data rate mobility-aware AODV-based protocol for flying ad-hoc networks," *Elsevier Veh. Commun.*, vol. 18, Aug. 2019, Art. no. 100163.

[13] J. Jiang and G. Han, "Routing protocols for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 58–63, Jan. 2018.

[14] O. Oubbati, M. Atiquzzaman, P. Lorenz, M. Tareque, and M. Hossain, "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81 057–81 105, Jun. 2019.

[15] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.1 1 b for "Automobile" Users," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 362–373.

[16] Y. Zhang, J. Zhao, and G. Cao, "Service scheduling of vehicle-roadside data access," *Springer Mobile Netw. Appl.*, vol. 15, no. 1, pp. 83–96, Feb. 2010.

[17] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method–PEPA," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 745–748, Apr. 2017.

[18] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel prediction based scheduling for data dissemination in VANETs," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1409–1412, Jun. 2017.

[19] K. Bok, J. Lim, S. Hong, and J. Yoo, "A multiple RSU collaborative scheduling scheme for data services in vehicular ad hoc networks," *Cluster Comput.*, vol. 20, no. 2, pp. 1167–1178, Jun. 2017.

[20] K. Liu, J. Ng, V. Lee, S. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a software defined network," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1759–1773, Jun. 2016.

[21] R. Zhang, X. Cheng, L. Yang, X. Shen, and B. Jiao, "A novel centralized TDMA-based scheduling protocol for vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 411–416, Feb. 2015.

[22] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, "A green strategic activity scheduling for UAV networks: A sub-modular game perspective," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 58–64, May 2016.

[23] K. Li, W. Ni, X. Wang, R. Liu, S. Kanhere, and S. Jha, "EPLA: Energy-balancing packets scheduling for airborne relaying networks," in *Proc. IEEE ICC*, Jun. 2015, pp. 6246–6251.

[24] J. Li, Y. Zhou, L. Lamont, and M. Déziel, "A token circulation scheme for code assignment and cooperative transmission scheduling in CDMA-based UAV ad hoc networks," *Wireless Netw.*, vol. 19, no. 6, pp. 1469–1484, Aug. 2013.

[25] C. Tipantuña, X. Hesselbach, V. Sánchez-Aguero, F. Valera, I. Vidal, and B. Nogales, "An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles," *Wireless Commun. Mobile Comput.*, vol. 2019, Feb. 2019, Art. no. 4734821.

[26] N. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[27] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, Nov. 2015.

[28] T. Long, M. Ozger, O. Cetinkaya, and O. Akan, "Energy neutral Internet of Drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, Jan. 2018.

[29] C. Pu, S. Lim, B. Jung, and J. Chae, "EYES: Mitigating forwarding misbehavior in energy harvesting motivated networks," *Elsevier Comput. Commun.*, vol. 124, pp. 17–30, Jun. 2018.

[30] M. Asadpour *et al.*, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, Mar. 2017.

[31] K. Yoon and C. Hwang, *Multiple Attribute Decision Making: An Introduction*. Newbury Park, CA, USA: Sage, 1995.

[32] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[33] C. Pu, S. Lim, J. Chae, and B. Jung, "Active detection in mitigating routing misbehavior for MANETs," *Wireless Netw.*, vol. 25, no. 4, pp. 1669–1683, May 2019.

[34] *Intel Pentium 4 Processor Datasheet*. [Online]. Available: http://download.intel.com/design/Pentium4/datashts/29864312.pdf, Accessed on: Feb, 2005.

Cong Pu (Member, IEEE) is currently an Assistant Professor with the Department of Computer Sciences and Electrical Engineering, Marshall University, West Virginia, USA. His research interests include blockchain, cybersecurity, Internet of Things, wireless networks, mobile computing, and information-centric networking.

Logan Carpenter is currently an undergraduate student with the Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA. His research interests include wireless networks and mobile computing.