Green Data-Collection From Geo-Distributed IoT Networks Through Low-Earth-Orbit Satellites

Huawei Huang, *Member, IEEE*, Song Guo, *Senior Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, Kun Wang, *Senior Member, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

***Abstract*—As a critical supplementary to terrestrial communication networks, low-Earth-orbit (LEO) satellite-based communication networks have been gaining growing attention in recent years. In this paper, we focus on data collection from geo-distributed Internet-of-Things (IoT) networks via LEO satellites. Normally, the power supply in IoT data-gathering gateways is a bottleneck resource that constrains the overall amount of data upload. Thus, the challenge is how to collect the data from IoT gateways through LEO satellites under time-varying uplinks in an energy-efficient way. To address this problem, we first formulate a novel optimization problem, and then propose an online algorithm based on Lyapunov optimization theory to aid green data-upload for geo-distributed IoT networks. The proposed approach is to jointly maximize the overall amount of data uploaded and minimize the energy consumption, while maintaining the queue stability even without the knowledge of arrival data at IoT gateways. We finally evaluate the performance of the proposed algorithm through simulations using both real-world and synthetic data traces. Simulation results demonstrate that the proposed approach can achieve high efficiency on energy consumption and significantly reduce queue backlogs compared with an offline formulation and a greedy "Big-Backlog-First" algorithm.**

***Index Terms*—Green data-collection, LEO satellite, Internet-of-Things (IoT).**

I. Introduction

INTERNET-OF-THINGS (IoT) networks have been widely applied to various applications, such as the remote surveillance systems used to monitor natural disasters, wild animals and environmental parameters of climate change, as well as the precision agriculture and other remote asset-management networks shown in Fig. 1.

Manuscript received August 23, 2018; revised January 11, 2019; accepted March 24, 2019. Date of publication April 4, 2019; date of current version August 16, 2019. This work was supported in part by NSFC under Grant 61872195, 61872310, and in part by the Shenzhen Basic Research Funding Scheme under Grant JCYJ20170818103849343. The associate editor coordinating the review of this paper and approving it for publication was E. Ayanoglu. *(Corresponding author: Huawei Huang.)*

H. Huang is with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China (e-mail: huanghw28@mail.sysu.edu.cn).

S. Guo is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: song.guo@polyu.edu.hk).

W. Liang is with the Research School of Computer Science, Australian National University, Canberra, ACT 2601, Australia (e-mail: wliang@cs.anu.edu.au).

K. Wang is with the Department of Electrical and Computer Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: wangk@ucla.edu).

A. Y. Zomaya is with the School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia (e-mail: albert.zomaya@sydney.edu.au).

Digital Object Identifier 10.1109/TGCN.2019.2909140

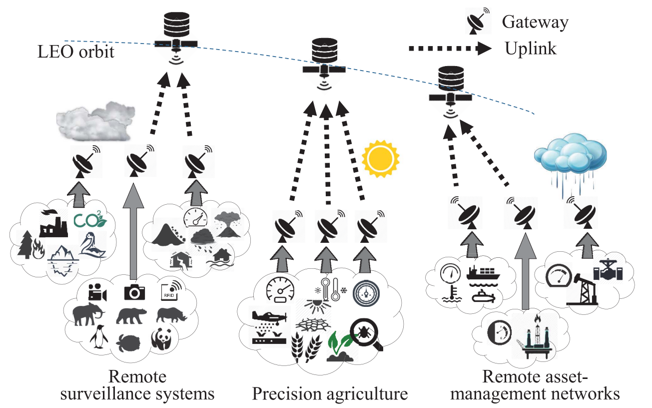


Fig. 1. Data collection from geo-distributed IoT networks via LEO satellites. Weather conditions greatly affect the channel state of uplinks.

Tremendous numbers of IoT devices and data-gathering gateways in the edge together constitute the data-sensing and capturing system. The data-sensing devices may have low cost and long battery lives based on the emerging Narrow-band IoT technology [1]. In large-scale geo-distributed IoT networks, such as oil & gas platforms located in remote locations, data-sensing can be accomplished by well-connected ground IoT networks. However, the problem is how to timely and efficiently gather data cached in distributed IoT gateways, and then forward the data to data centers for further analysis.

For urban IoT networks, some existing studies [2]–[4] use the cellular networks such as 3G, 4G or potentially 5G technologies to establish dedicated data gathering networks. For the offshore IoT networks, studies [5], [6] explore the use of UAVs to gather data sensing from offshore ocean-observation devices. However, these approaches are technically impossible or prohibitive in terms of their operation cost for large-scale geo-distributed IoT networks. Recently, low-earth-orbit (LEO) satellite based constellation networks have been launched and LEO satellite based projects, e.g., *OneWeb*, *SpaceX*, and *Boeing*, announced to provide global Internet-access services. Under the fully covered global access networks [7], [8], LEO satellites provide great opportunities to the geo-distributed IoT networks. However, the challenge is to design energy-efficient data gathering schemes to aggregate the data caching in IoT gateways under the LEO satellite based access networks.

Adopting this new data gathering scheme is based on the following three aspects. First, the power supply for the large number of IoT gateways isolated in remote locations is viewed as a bottleneck constraint [6]. Second, the uplinks from IoT gateways to LEO satellites are time-varying dynamic channels, which are particularly sensitive to weather conditions. For example, as shown in Fig. 1, the weather conditions usually differ from gateway to gateway. Transmitting the same volume of data under a bad channel condition consumes much more energy than that under a good condition [9]. Finally, if the data cached in IoT gateways are not gathered in a timely manner, the successive data stream will flush them shortly. The so-called *buffer overflow* problem [10]–[12] will incur data loss. Therefore, it is significant to design an optimal scheduling mechanism for online data collection from geo-distributed IoT networks such that the total energy consumption is minimized, the overall amount of data uploaded can be maximized, and the data overflow in gateways can be also avoided.

In this paper, a novel optimization problem based on this application scenario is formulated. Then, an online scheduling framework is developed using the Lyapunov optimization technique [13]. The main contributions of this paper are described as follows.

- We study a novel green online data gathering problem for the geo-distributed IoT networks using the LEO networks. The novelty of this problem relies in the consideration of the time-varying uplinks due to the relative motion between LEO satellites and IoT gateways.
- To jointly minimize energy consumption and maximize overall data uploaded, we devise a novel online algorithm for green data-uploading, which can avoid the buffer overflow problem during data gathering from the geo-distributed IoT networks as well. The theoretic characteristics of the online algorithm, such as the optimality gap and the stability of gateway queues, are analyzed rigorously.
- Finally, based on real-world traces of LEO constellation, the simulation results show that the proposed online algorithm achieves much higher efficiency of energy consumption and lower queue backlogs than a greedy "Big-Backlog-First" algorithm.

The rest of this paper is organized as follows. Section II reviews related work. Section III specifies system model and problem formulation. Section IV presents the proposed online scheduling framework. Section V conducts performance evaluation. Finally, Section VI concludes the paper.

II. Related Work

In the perspective of data gathering for IoT networks, various approaches have been proposed for different scenarios. For example, Barbatei *et al.* [5] presented a UAV based prototype that can gather and relay data from the sensor nodes deployed in remote areas or floating on water surface. Zolich *et al.* [6] combined the UAV and the low-cost buoys hardware to implement a sensor data collection system, which has been used to gather the underwater sensor data in Norwegian subarctic fjord. To enable the IoT data collection processes for multiple parties, Cheng *et al.* [14] made use of a concurrent data collection tree to improve the collection effectiveness of IoT applications. A mobile satellite communication services company *Isatdata Pro* [8] exploited the LEO satellites to provide the global communication services for Machine-to-Machine (M2M) applications. This is very useful to relay the sensor data from remote assets such as oil, gas, maritime, commercial fishing and heave equipment sectors.

Several studies related to satellite based communication networks have been recently conducted. For example, Wu *et al.* [15] proposed a two-layer caching model for content delivery services in satellite-terrestrial networks. Jia *et al.* [16] studied data transmission and downloading by exploiting the inter-satellite links in the LEO satellite based communication networks. Cello *et al.* [17] proposed a selection algorithm to mitigate network congestion, using the nano-satellites in the LEO based networks.

Comparing with existing studies, we particularly focus on green online data gathering problem from the global distributed IoT networks using LEO satellites.

In an earlier version of this work [18], we have studied a basic online data gathering problem for geo-distributed remote IoT networks. In contrast, we further consider the stability of gateway queues in the problem formulation of this paper. We also provide theoretic analysis on the optimality gap of the new online algorithm while considering the queue stability. In another article [19], we study a problem contrary to that of this paper, i.e., how to download data from the LEO satellite based datacenter in an energy-efficient manner.

III. Problem Formulation

A. System Model

We consider a discrete-time system measured in time slots $t \in \{1, 2, ... T\}$, where T denotes the number of time slots. The length of each slot is denoted by δ, which ranges from hundreds of milliseconds to seconds [20]. We then focus on geo-distributed IoT networks $\mathcal{G} = \langle I \cup J, E(t) \rangle$, where I and J are a set of ground IoT data-gathering gateways and LEO satellites orbiting in specific planes, respectively. $E(t)$ is a set of time-varying uplinks in time slot t between the IoT data-gateways and LEO satellites. The gathered data can be temporally stored in satellites and transmitted to ground stations eventually. Note that, we only study data gathering through uplinks in this paper.

Since LEO satellites are orbiting in their planes according to predefined parameters, the time-varying available uplinks between the ground gateways and satellites can be known as a priori in each time slot. We use $(i, j) \in E(t)$ to denote an uplink channel between an IoT gateway $i \in I$ and a LEO satellite station $j \in J$, and let c_{ij}^t represent the channel state of (i, j) at time slot t. The time-varying channel state can be obtained by direct measurement [9] or by prediction [21]. We thus assume that the channel state can be known by the system controller at the beginning of a time slot in our system model. Every satellite has a data-receiving rate capacity, which is denoted by C_j, $j \in J$.

In the geo-distributed IoT network scenario, we consider the power supply as the bottleneck resource [7] in IoT gateways, rather than the frequency resource in LEO satellites, because modern high-throughput satellites can achieve high transmission capability using the technology of frequency reuse in multiple spot beams. For our system model, the frequency bandwidth for satellite uplinks is first divided into a group of orthogonal narrow channels exploiting the Orthogonal Frequency Division Multiplexing (OFDM) technology [22], [23]. When multiple gateways connect to the same LEO satellite, we assume that the gateways are using uploading channels under a combination of Frequency Division Multiple Access (FDMA) [24] and Time Division Multiple Access (TDMA) [24] techniques. Under such a hybrid mechanism, each IoT gateway is assigned a unique channel to one of its available uplinks during the specified time slots for its data uploading. To avoid data uploading overlapping, multiple gateways can connect to the same LEO satellite by using either (i) a same transmission channel at different time slots, or (ii) different transmission channels at a same time slot, and subject to the constraint of the satellite's data-receiving rate capacity. In other words, a transmission channel can be reused by different uplinks at different time slots, and different gateways must use different channels to connect to the same satellite at the same time slot.

On the other hand, if an uplink is configured for a gateway, it can be served immediately to upload packets to the associated satellite. Taking both the division multiple access mechanism and the dynamic gateway-to-satellite contact windows into account, we consider the *preemptive* model for each time-varying uplink. If an uplink is called *preemptive*, a data-uploading task conducting through this uplink in a time slot can be replaced by another uploading task in the next time slot, according to a predefined priority policy of channel allocation.

We then describe the relationship between the power allocation and transmission rate on an uplink by referring to a well-adopted concave rate-power curve $g(p, c)$ [9], [25] as shown in Fig. 2(a), where p and c denote the power allocation and the channel condition, respectively. The maximum transmission rate of each uplink is denoted as $\mu_{\max}$ under arbitrary channel conditions, i.e., $g(p,c) \leq \mu_{\max}, \forall p \in \overrightarrow{P}, \forall c \in \overrightarrow{C}$, where $\overrightarrow{C}$ is a vector of given channel conditions.

In practice, the power-allocation parameter in a transmitter adopts linear piecewise power-rate curves with a pre-defined finite set of discrete operating *gears* [9], [25] denoted by $\overrightarrow{P} = [p_1, p_2, \ldots, p_{\max}]$, rather than a continuous concave function as shown in Figure 2(a). Thus, the transmission rate of an uplink is determined by two critical parameters, i.e., the power gear allocated and the currently observed channel condition.

As shown in Fig. 2(b), the volume of IoT data stream arriving to each gateway $i \in I$ at each time slot t is denoted as $a_i(t)$. Note that, we assume all the data-arrival rates at IoT gateways are within a positive peak value R_{max}. Let $Q_i(t)$ denote the time-varying backlog of the queue residing in gateway i. It can be seen that $Q_i(t)$ keeps growing if the data in gateway i cannot be successfully collected by satellites, and finally triggers buffer overflow in the gateway.

Important notations are also explained in Table I.

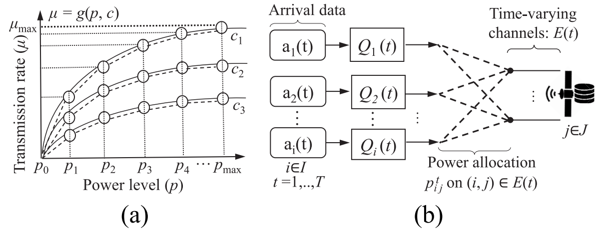


Fig. 2. (a) Shows the classical Shannon's Theorem based piecewise rate-power curve [9], [25] with parameters: power-supply gear p and channel condition c. (b) Illustrates the system model.

TABLE I
NOTATIONS AND VARIABLES

I	the set of IoT gateways
J	the set of LEO satellites
$E(t)$	the set of time-varying uplinks available in time slot t between LEO satellites and IoT gateways
T	the number of time slots
δ	the length of a time slot
R_{max}	peak value of data-arrival rate at gateways
$a_i(t)$	the volume of IoT data received at gateway $i \in I$ in the beginning of time slot t
$Q_i(t)$	the time-varying queue backlog at gateway $i \in I$
(i,j)	$\in E(t)$, an uplink from $i \in I$ to $j \in J$
$\overrightarrow{P}$	vector of power gears
$\overrightarrow{C}$	vector of channel conditions
$g(p,c)$	transmission-rate function of uplinks, with power gear $p \in \overrightarrow{P}$ and channel state $c \in \overrightarrow{C}$
p_{ij}^t	$\in \overrightarrow{P}$, the variable indicating the power allocated on the uplink $(i,j) \in E(t)$

B. Problem Statement and Formulation

1) Variables: Given the system model described above, the crucial control decision we need to make is the power allocation for each uplink channel. Therefore, we define a real-valued variable $p_{ij}^t \in \overrightarrow{P}$ to represent the power allocation level on the uplink $(i,j) \in E(t)$ at time slot t.

2) Performance Metrics: For data collection from the geo-distributed IoT networks, the overall amount of data uploaded is the most critical performance metric, which should be devoted to improve. Denoted by *data*(t), we define the time-varying *data amount* uploaded at time slot t as

$$data(t) = \sum_{(i,j) \in E(t)} g\left(p_{ij}^t, c_{ij}^t\right) \cdot \delta, \forall t. \quad (1)$$

As mentioned earlier, the data-upload in an IoT gateway is constrained by its energy-budget. If the power allocation on uplink channels cannot be carefully scheduled, e.g., allocating too large power gear to an uplink with bad channel condition, much energy is going to be wasted, thus reducing the overall amount of data uploaded. Therefore, the energy consumption should be minimized when uploading data to satellites. Denoted by *eng*(t), the total *energy consumption* spent on data-uploading throughout all ground gateways at time slot t is calculated as

$$eng(t) = \sum_{(i,j) \in E(t)} \delta \cdot p_{ij}^t, \forall t. \quad (2)$$

To maximize the overall amount of data uploaded and minimize the energy consumption simultaneously, we define a penalty function that positively associates with the numerical energy consumption *eng*(*t*) and reversely associates with the numerical data amount *data*(*t*). The objective is to minimize a time-average penalty, which is denoted by $\overline{Pen}$, while all queue backlogs are keeping mean-rate stable. Note that, we call a queue in gateway $i \in I$ is *mean-rate stable* [13], if it satisfies $\lim\limits_{t\to\infty} \frac{E\{Q_i(t)\}}{t} = 0$. We thus have the following penalty-minimization formulation.

$$\min \ \overline{Pen} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} [\beta \cdot eng(t) - data(t)] \tag{3}$$

$$\text{s.t.} \sum_{\substack{(i,j')\in E(t),\\ \&(j'=j)}} g\left(p_{ij}^t, c_{ij}^t\right) \le C_j, \ c_{ij}^t \in \overrightarrow{C}, \forall j \in J \tag{4}$$

$$Q_i(t) \text{ is mean-rate stable}, \ \forall i \in I$$
$$\text{Variables: } p_{ij}^t \in \overrightarrow{P}, \forall (i,j) \in E(t), \forall t = 1, \ldots, T. \tag{5}$$

In the objective function (3), β indicates the weight of energy consumption in the penalty function. By tuning β, we have an integrated numerical objective in (3). Let C_j denote the total data receiving rate capacity of LEO satellite $j \in J$ at any time slot, inequality (4) indicates that the total uploading data rate should not exceed the capability of each satellite when it is receiving data from ground IoT gateways. Finally, constraint (5) ensures the stability in gateway queues.

IV. Online Scheduling Framework

In this section, we strive for a near-optimal solution to the online green data gathering problem (3) using the queue backlog theory under the Lyapunov optimization framework [13]. The Lyapunov optimization technique is a kind of *stochastic optimization*, which can be used to address the online control problems by manipulating the queue backlogs in system. Under this framework, queue backlogs are extremely useful for designing dynamic algorithms that do not require a-priori knowledge of channel statistics.

A. Problem Transformation

1) Dynamics of Queues: Recall that the backlog $Q_i(t)$ represents the data size measured in bits in the queue of gateway $i \in I$. A small backlog indicates queue stability, while a large one implies high probability of buffer overflow. Initially, $Q_i(1) = 0, \forall i \in I$. Afterwards, the time-varying queue backlog of each IoT gateway evolves as follows.

$$Q_i(t+1) = \max\left[Q_i(t) - b_i(t), 0\right] + a_i(t), \forall i \in I, \tag{6}$$

where $b_i(t) = \delta g(p_{ij}^t, c_{ij}^t), (i,j) \in E(t)$, represents the total diminishing bits of the backlog Q_i.

2) Virtual Queues: We then transform the original minimization problem (3) into a pure queue-stability problem based on Lyapunov optimization theory [13]. To make sure the constraint (4) still holds, we define a virtual queue X_j for each satellite $j \in J$ with the following update function.

$$X_j(t+1) = \max\left[X_j(t) + x_j(t), 0\right], \ \forall t = 1, \ldots, T, \tag{7}$$

where $x_j(t) = \sum_{i:(i,j)\in E(t)} g(p_{ij}^t, c_{ij}^t) - C_j$, $\forall j \in J$; $\forall t = 1, \ldots, T$. The initial backlog is $X_j(1) = 0$ for each virtual queue.

Insight: By summing $X_j(t)$ over time slots $t = 1, \ldots, T$, we have $\frac{X_j(T)}{T} - \frac{X_j(1)}{T} \ge \frac{1}{T}\sum_1^T x_j(t)$. With $X_j(1) = 0$, take expectations on both sides and let $T \to \infty$, we get $\lim_{T\to\infty} \sup \frac{\mathbb{E}\{X_j(T)\}}{T} \ge \lim_{T\to\infty} \sup \overline{x_j}(t)$, where $\overline{x_j}(t)$ is the time-average expectation of $x_j(t)$ over $t = 1, \ldots, T$. If $X_j(t)$ is mean-rate stable [9], we have $\lim_{T\to\infty} \sup \frac{\mathbb{E}\{X_j(T)\}}{T} = 0$, which indicates that $\lim_{T\to\infty} \sup \overline{x_j}(t) \le 0$. This implies that the desired constraints for $x_j(t)$ are satisfied.

Then, combining all actual and virtual queues, we can obtain a concatenated vector $\Theta(t) = [\mathbf{Q}(t), \mathbf{X}(t)]$ with update equations (6) and (7). Next, a Lyapunov function of the geo-distributed data gathering system is defined as follows.

$$L(\Theta(t)) \triangleq \frac{1}{2}\sum_{i\in I} Q_i(t)^2 + \frac{1}{2}\sum_{j\in J} X_j(t)^2. \tag{8}$$

In fact, $L(\Theta(t))$ calculates a scalar volume of queue congestion [13] in the geo-distributed data gathering system. Normally, a Lyapunov function with a small value indicates short backlogs of both actual and virtual queues. Thus, the system could keep in a stable state.

3) Drift-Plus-Penalty Expression: We then define a *one-slot conditional Lyapunov drift* [13], denoted by $\Delta(\Theta(t))$, which is calculated as

$$\Delta(\Theta(t)) = \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)\}. \tag{9}$$

Insight: Given the current backlogs of the system $\Theta(t)$, the drift shown as equation (9) depicts the expectation of variation measured in Lyapunov function (8) over one time slot. Under the framework of Lyapunov optimization, the supremum bound of Lyapunov drift-plus-penalty expression is expected to be minimized in each time slot, aiming to retrieve the near-optimal decisions for our proposed original green data gathering problem.

Thus, the transformed problem is rewritten as the follows.

$$\begin{aligned} \min \ & \Delta(\Theta(t)) + V\mathbb{E}\{\beta \cdot eng(t) - data(t) | \Theta(t)\} \\ \text{s.t.} \ & p_{ij}^t \in \overrightarrow{P}, \forall t = 1, \ldots, T. \end{aligned} \tag{10}$$

In (10), *V* is a tunable knob denoting the weight of penalty. The objective function (10) reaffirms our three-fold goals for the online green data gathering from geo-distributed IoT networks: (1) to minimize the energy consumption, (2) to maximize the overall amount of data uploaded, and (3) to maintain the stability of the holistic system meanwhile.

We then have the following theorem.

Theorem 1: Given that the data arrival rate $a_i(t)$, the time-varying available uplink set $E(t)$, the backlogs of both actual and virtual queues are observable at each slot *t*, for any value of $\Theta(t)$, the Lyapunov drift $\Delta(\Theta(t))$ of the geo-distributed IoT data gathering system under arbitrary control policies satisfies the following results:

$$\Delta(\Theta(t)) \leq B + \sum_{i \in I} Q_i(t)\mathbb{E}\{a_i(t) - b_i(t)|\Theta(t)\} + \sum_{j \in J} X_j(t)\mathbb{E}\{x_j(t)|\Theta(t)\}, \tag{11}$$

where $B = \frac{1}{2}|I|[R_{max}^2 + (|J| + \delta^2)\mu_{\max}^2] + \sum_{j \in J} C_j(\frac{1}{2}C_j - |I|\mu_{max})$ is a positive constant. Note that, $|.|$ represents the size of a set.

Please find the poof of Theorem 1 from the Appendix-A of our online technical report [26]. Based on Theorem 1, we then derive the upper bound of *drift-plus-penalty* expression for the geo-distributed data gathering system by combining (10) and (11) as follows.

$$\Delta(\Theta(t)) + V\mathbb{E}\{\beta \cdot eng(t) - data(t)|\Theta(t)\} \leq B + V\delta \sum_{(i,j) \in E(t)} \left[\beta p_{ij}^t - g\left(p_{ij}^t, c_{ij}^t\right)\right] \tag{12}$$

$$+ \sum_{i \in I} Q_i(t)\mathbb{E}\left\{a_i(t) - \delta g\left(p_{ij}^t, c_{ij}^t\right)|\Theta(t)\right\} \tag{13}$$

$$+ \sum_{j \in J} X_j(t)\mathbb{E}\{x_j(t)|\Theta(t)\}. \tag{14}$$

B. Online Scheduling Algorithm

Unlike existing offline solutions that make decisions based on the known data-arriving rates, we do not make such an impractical assumption. Instead, we design our online scheduling algorithm only depending on the observed queue backlogs in each time slot. Driven by the upper bound of *drift-plus-penalty* expression derived in the end of last subsection, it can be seen that minimizing the objective in (10) is equivalent to minimizing expressions (12), (13) and (14) jointly. Thus, we have proposed a two-phase online data-gathering Algorithm 1.

1) Phase-I, Power Allocation on Uplinks: In each time slot, the power allocation decisions on uplinks are independent among different gateways. Therefore, the power allocation can be accomplished by the centralized system controller for each individual gateways without having to know the backlog information from other gateways. This is a very practical merit for the large-scale global geo-distributed IoT networks.

Let (p, c) be short for the term (p_{ij}^t, c_{ij}^t), we have the following subproblem (15):

$$\begin{aligned} &\min \ \Gamma(p, c) \\ &\text{s.t.} \ \ p_{ij}^t \in \overrightarrow{P}, (i,j) \in E(t), i \in I, \forall t, \end{aligned} \tag{15}$$

where $\Gamma(p, c) = V[\beta p_{ij}^t - g(p_{ij}^t, c_{ij}^t)] - Q_i(t)g(p_{ij}^t, c_{ij}^t) + X_j(t)g(p_{ij}^t, c_{ij}^t)$, $c_{ij}^t \in \overrightarrow{C}$.

It can be observed that the problem (15) is a linear programming. Partially differentiating $\Gamma(p, c)$ with respect to p and rearranging terms, we have

$$\frac{\partial \Gamma(p, c)}{\partial p} = V\beta + \left[X_j(t) - Q_i(t) - V\right]\frac{\partial g(p, c)}{\partial p}. \tag{16}$$

Algorithm 1: Online Green Data-Gathering

Input : $E(t)$, observed time-varying queue backlogs and channel conditions
Output: power gears $p_{ij}^t \in \overrightarrow{P}, (i,j) \in E(t), \forall t$

while *in each time slot t* **do**
  Phase-I: allocate power on uplinks:
  for *each* $(i,j) \in E(t)$ **do**
    allocate a power gear for uplink (i, j), according to equation (19).
  Phase-II: Update $\mathbf{Q}(t)$ and $\mathbf{X}(t)$ by invoking equation (6) and equation (7), respectively.

Note that, the term $\frac{\partial g(p,c)}{\partial p}$ in each discrete power supply level $p \in \overrightarrow{P}$ can be easily retrieved under the observed channel condition c. Let p vary within the vector $\overrightarrow{P} = [p_1, p_2, \ldots, p_{\max}]$, a vector of derivative values can be obtained as follows.

$$\overrightarrow{\mathbb{D}} = \left[\frac{\partial \Gamma(p, c)}{\partial p_1}, \frac{\partial \Gamma(p, c)}{\partial p_2}, \ldots, \frac{\partial \Gamma(p, c)}{\partial p_{\max}}\right]. \tag{17}$$

Since $g(p, c)$ is a concave function, which determines that $\Gamma(p, c)$ is convex. By equation (16), we have the valley point (p^*, c_{ij}^t) of $\Gamma(p, c)$ such that

$$\frac{\partial g\left(p^*, c_{ij}^t\right)}{\partial p^*} = \frac{V\beta + X_j(t)}{Q_i(t) + V - X_j(t)}. \tag{18}$$

Finally, the power-allocation solution can be chosen from the given power-level vector as follows.

$$p_{ij}^t = \begin{cases} p_{\min}, & \text{if elements (ele.) in } \overrightarrow{\mathbb{D}} \text{ are non-negative;} \\ p_{\max}, & \text{if ele. in } \overrightarrow{\mathbb{D}} \text{ are non-positive;} \\ p^- \text{ or } p^+: & \arg\min\left\{\Gamma\left(p^-, c_{ij}^t\right), \Gamma\left(p^+, c_{ij}^t\right)\right\}, \text{ if ele.} \\ & \text{in } \overrightarrow{\mathbb{D}} \text{ vary from negative to positive,} \end{cases} \tag{19}$$

where p^- and p^+ are two successive discrete power gears such that $p^- \leq p^* \leq p^+$, where $p^-, p^+ \in \overrightarrow{P}$, and p^* is the optimal power gear denoted by the valley point (p^*, c_{ij}^t).

2) Phase-II, Queue Update: In the end of each time slot, using the optimal solutions p_{ij}^t, the actual queues $\mathbf{Q}(t)$ and the virtual queues $\mathbf{X}(t)$ need to be updated by invoking equation (6) and equation (7), respectively.

C. Optimality Gap and Stability of Gateway Queue

We now show the optimality and the queue-stability of the devised online scheduling algorithm.

Theorem 2: For arbitrary data arrival rate $a_i(t) \leq R_{max}$, $i \in I, \forall t$, the proposed online scheduling algorithm can yield a solution ensuring that:

(a) the gap between the achieved time-average penalty and the optimal one Pen^{opt} is within $\frac{B}{V}$, i.e.,

$$\lim_{T \to \infty} \sup \frac{1}{T} \sum_{t=1}^{T} \{\beta \cdot eng(t) - data(t)\} \leq Pen^{opt} + \frac{B}{V}, \tag{20}$$

where $Pen^{opt} = \lim_{T\to\infty} \inf \frac{1}{T}\sum_{t=1}^{T}\{\beta\overline{e}(t) - \overline{d}(t)\}$, $\overline{e}(t)$ and $\overline{d}(t)$ are the resulted energy-consumption and the overall amount of uploaded data indicated by the optimal solution to the optimization (3);

(b) all the queues in the uploading gateways are mean-rate stable.

Please refer the Appendix-B of our online technical report [26] for the poof of Theorem 2.

D. Simulation Settings

1) Basic Settings: The performance of the proposed online green data gathering algorithm is evaluated using the well-known emulator Satellite Tool Kit (STK) [27], which is designed by AGI (Analytical Graphics, Inc.). It is a useful analytical tool offering scientists and engineers the strong capability to analyze complex datasets such as terrestrial, oceanic and aerial assets. Using STK, we retrieve the contact trace between LEO satellites and the terrestrial IoT gateways at different time slots. To strengthen the simulation, we build a LEO system based on the widely-adopted Globalstar constellation [16], [28], which is composed of 48 LEO satellites averagely distributed in 8 orbital planes.

In total 216 IoT gateways are deployed globally and averagely in the world map. We also generate the synthetic channel-state traces with three states (i.e., good, medium and bad) according to weather conditions of all locations obtained from the Internet. The one-day mission of LEO satellites starts from 12 July 2017 00:00:00 UTCG (Gregorian Coordinated Universal Time). The length of each time slot is set as 10 seconds. The contact trace between each satellite and each IoT gateway is retrieved at each time slot.

V. Performance Evaluation

On the other hand, the bandwidth of each uplink channel is set to 1 megahertz (MHz). To calculate the data-receiving rate of uplinks, we adopt the classic Shannon's Theorem based rate-power function [25]:

$$g(p, c) = bw \cdot \log(1 + \upsilon(c) \cdot p), \tag{21}$$

where the bandwidth $bw = 1$ MHz, and $\upsilon(c)$ determines the fading coefficient depending on the channel state c. As the three-state condition model [21] adopted to depicts the satellite channels, $\upsilon(c)$ is equal to 5.03, 3.46 and 1.0 corresponding to the *good*, *medium* and *bad* conditions. The power-level vector $\overrightarrow{P}$ is set to 11 gears averagely varying from 0 Watt to 1 Watt. We then generate the synthetic data-arrival traces for each IoT gateways with the predefined range, denoted by α_{LB} and α_{UB}, of the arrival data-volume in each time slot. In simulation, we set α_{LB} and α_{UB} to 10 Megabits (Mbits) and 100 Mbits, respectively. In addition, β and V are both set to 1.0 by default unless otherwise is claimed.

2) Metrics: We evaluate the performance of the proposed online algorithm with five metrics: overall amount of data uploaded (measured by *bits*), total energy consumption (measured by *Watt·second*, w·s for short), numerical penalty, efficiency of energy consumption and queue backlogs(measured by *bits*). Particularly, the *efficiency of energy consumption* is inversely associated with the numerical energy consumption spending on uploading per bit of data, denoted by *Watt·second per bit* or *Watt·sec/bit* hereafter. The insight we design this metric is that a good algorithm probably yields both a higher overall amount of data uploaded and a larger energy consumption than a worse algorithm do. Therefore, the most fair way to evaluate the performance of algorithms is the efficiency of energy consumption used in the data-uploading through uplinks. Finally, the queue-backlog is the indicator of the system stability. Thus, backlog should be made as small as possible in each queue.

Algorithm 2: Big-Backlog-First (**BBF**) (ζ)

Input : ζ, $E(t)$, observed time-varying queue backlogs and channel conditions

Output: power gears $p_{ij}^{t} \in \overrightarrow{P}, (i, j) \in E(t), \forall t$

```
while in each time slot t do
    for satellite j' ∈ J do
        π ← Sort all the gateway located in the coverage
            of satellite j' in a non-increasing order by their
            queue backlogs.
        for gateway i' ∈ π do
            while the data-receiving capacity of satellite
                  j' is conserved do
                Allocate a power gear for uplink (i', j'),
                according to equation (21), to reduce the
                percentage of backlog in i' by ζ.
```

A. Benchmark Schemes

1) A Variant Offline Formulation: It should be noted that there is not exactly the same offline formulation corresponding to the proposed online scheduling problem. However, we still study the most similar variant offline version of the proposed online scheduling problem. Different from the online formulation (3)-(5), the following offline formulation is constructed using integer linear programming (ILP) techniques, provided that the arriving data in gateways is known within an optimization window *T*.

In particular, we define a binary variable p_{ij}^{ty} to denote whether to assign the power gear $y \in \overrightarrow{P}$ for uplink $(i, j) \in E(t)$, i.e., $p_{ij}^{ty} = 1$ only if y is assigned to (i, j). Accordingly, the energy consumption at time slot t is recalculated as $eng(t) = \sum_{y\in\overrightarrow{P}}\sum_{(i,j)\in E(t)} \delta \cdot p_{ij}^{ty} \cdot y$, while the overall amount of data uploaded is recalculated as $data(t) = \sum_{y\in\overrightarrow{P}}\sum_{(i,j)\in E(t)} g(y, c_{ij}^{t}) \cdot p_{ij}^{ty} \cdot \delta$ for the time slot $t = 1, \ldots, T$.

$$\min \ \overline{Pen} = \frac{1}{T}\sum_{t=1}^{T}[\beta \cdot eng(t) - data(t)] \tag{22}$$

$$\text{s.t.} \sum_{\substack{(i,j')\in E(t),\\ \& j'=j}} \sum_{y\in\overrightarrow{P}} g\left(y, c_{ij}^{t}\right) \cdot p_{ij}^{ty} \le C_j, \ \forall j \in J \tag{23}$$

$$\sum_{y\in\overrightarrow{P}} p_{ij}^{ty} \leq 1,\ \forall(i,j)\in E(t) \tag{24}$$

$$Q_i(t+1) \leq G_i,\ \forall i\in I, \forall t=1,\ldots,T-1.$$

$$\text{Var:}\ \ p_{ij}^{ty}\in\{0,1\}, y\in\overrightarrow{P}, \forall(i,j)\in E(t), \forall t=1,\ldots,T. \tag{25}$$

In this offline formulation depicted from (22) to (25), constraint (24) implies that the total number of power gears assigned to each uplink should be at most 1. Constraint (25) enforces that the queue backlog of each gateway should be limited by the backlog capacity $G_i, i\in I$. Notice that, we can still compute the $Q_i(t+1)$ by referring to (6). However, the diminishing bits of queue backlog in Q_i should be changed to $b_i(t)=\sum_{(i',j)\in E(t)\&(i'=i)} g(y,c_{ij}^t)\cdot p_{ij}^{ty}\cdot\delta, \forall i\in I$.

Compared with our online formulation, the major difference relies in constraint (25), in which we enforce each gateway a stringent capacity measured in the maximum backlog size, i.e., $G_i, \forall i\in I$. Notice that, the offline formulation will become *unbounded* if without specifying constraint (25). In contrast, we don't set a backlog capacity in our online approach, because we would like to achieve the queue stability even if without enforcing a capacity constraint on the gateways while the system keeps running in a long run. Thus, our proposed online approach is able to handle the highly dynamic communication networks. More importantly, the proposed online scheduling algorithm is a general approach, since it is adaptive to various data-collection IoT systems where the backlog capacity of gateways can be arbitrary.

2) BBF Algorithm: As another benchmark to compare the performance with the proposed one, we also devise a "Big-Backlog-First" (shorten as BBF) based algorithm 2. The basic idea includes the following two steps: (a) sort all the gateway queues in a non-increasing order by their queue backlogs; (b) only the first few gateways can use the time-varying available uplinks, while conserving the data-receiving rate capacity of each satellite. To guarantee the fairness when allocating power gears to the uplinks for the gateways that are first to be served, we define a normalized backlog-control parameter, denoted by ζ, which indicates the percentage of backlog that should be reduced in a selected queue through allocating power on the associated uplink.

B. Simulation Results

1) Comparison With Offline Performance: Even though the offline formulation is somewhat different from our online scheduling problem, we still compare their performance with respect to the overall amount of data uploaded, and the efficiency of each unit of consumed energy. We obtain the offline solutions and their corresponding performance using the popular ILP solver *Gruobi*, which has been widely adopted by both commercial usages and academia. In the simulation, the *Gruobi* solver can only solve a snapshot of network when the system keeps running within a specified optimization window. Taking the offline optimization needs a very long time to solve an optimization in a large network, we only

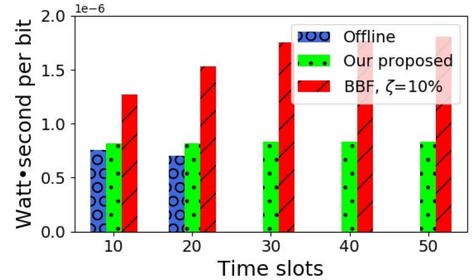


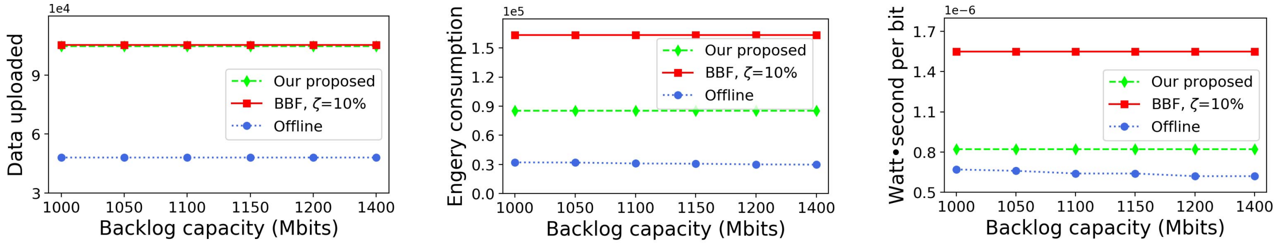
Fig. 3. Performance of "*Watt·second per bit*" under the offline scheme, comparing with the proposed online algorithm and the BBF algorithm (while $\zeta = 10\%$). Note that, the Offline scheme becomes *infeasible* when the optimization window exceeds the 20^{th} time slot.

deploy 108 gateways in all groups of simulation to evaluate the offline performance. Furthermore, to avoid the bias of optimization targets while applying the three schemes, we enforce $\beta=0$, which indicates that the optimization target is only to maximize the overall amount of data uploaded within the optimization windows.

In the first group of simulation, we compare the energy efficiency of the three schemes versus the varying optimization window measured from the 10^{th} to the 50^{th} time slots. Particularly, we set the backlog capacity of gateway to 1000 Mbits for the offline scheme, the backlog-control parameter (i.e., ζ) to 10% for the BBF algorithm, and the data-receiving rate capacity of satellites to 100 Mbits/s for all the three schemes. As shown in Figure 3, we can see that the offline scheme only yields feasible solutions for the first 20 time slots, because of the stringent constraint restricting the backlog capacity in each gateway. We can also observe that the unit energy consumed per bit, i.e., *Watt·second per bit*, of our proposed algorithm performs much lower than the BBF ($\zeta=10\%$), and slightly higher than the offline solutions under the current system settings.

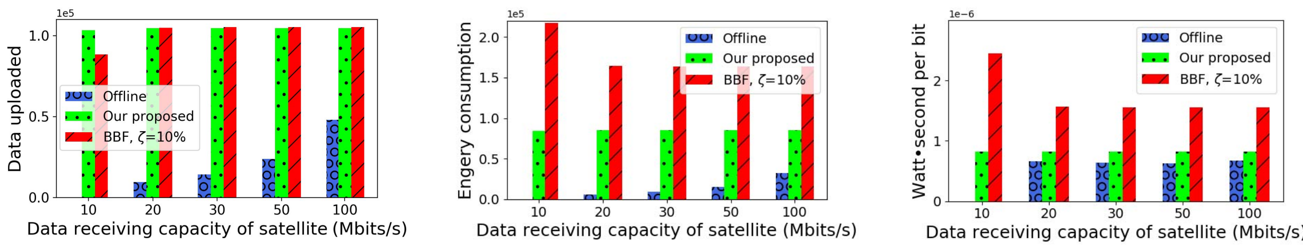
Although the proposed online algorithm and the BBF algorithm have no constraints on the backlog capacity of gateways, we still provide their performance showing in Fig. 4 as a comparison with the varying performance under the Offline scheme. By varying the backlog capacity of gateways within the range {1000, 1050, 1100, 1150, 1200, 1400} Mbits, we find that the overall amount of data uploaded showing in Fig. 4(a) has no changes. However, the energy consumption, showing in Fig. 4(b), reduces from 32028 Watt·sec to 29696 Watt·sec. Thus, the energy consumption spending on each unit uploaded data decreases from 0.00000067 Watt·sec/bit to 0.00000062 Watt·sec/bit, as shown in Fig. 4(c). This is because much more IoT data is allowed to queue in the gateways while the backlog capacity grows. Consequently, the offline scheme can find better transmitting opportunities under better weather conditions for the new queuing data. Thus, the energy efficiency can be improved.

We then evaluate the energy efficiency versus the data-receiving capacity of satellites by fixing T at 20 time slots, and assigning backlog-capacity-of gateways to 1000 Mbits/s for the offline scheme, while varying the data-receiving capacity of satellites within {10, 20, 30, 50, 100} Mbit/s. As



(a) (b) (c)

Fig. 4. Comparison with the Offline scheme while varying the backlog capacity of gateways. Note that, the efficiency of energy consumption reversely associates with the *Watt·second per bit* (*w·s/bit*). The data-receiving capacity of satellites is set to 100 Mbits/s. (a) Data uploaded (Mbits). (b) Energy consumption (*w·s*). (c) Watt·second per bit.



(a) (b) (c)

Fig. 5. Comparison with the Offline scheme while varying the data-receiving capacity of satellites. Note that, the backlog capacity of gateways are set to 1000 Mbits. Note that, the Offline scheme becomes *infeasible* when the data-receiving capacity of satellites is equal to 10 Mbits/s. (a) Data uploaded (Mbits). (b) Energy consumption (*w·s*). (c) Watt·second per bit.

shown in Fig. 5(a) and Fig. 5(b), both the overall amount of data uploaded and energy consumption are very low, because the Offline scheme can only find its unique optimal solutions restricted by the backlog capacity in gateways, i.e., 1000 Mbits. However, our proposed online algorithm and the BBF don't have such backlog capacity constraints, thus they yield both high overall amount of uploaded data and energy consumption. Therefore, as aforementioned, the only fair comparison is to measure their energy efficiency. From Fig. 5(c), we can see that the Offline scheme maintains the lowest *Watt·second per bit* performance, and our proposed online scheme has a slightly higher *Watt·second per bit* than the Offline optimal solution does.

Since the offline formulation is incapable to solve a large instance, we only compare the performance of the proposed online algorithm and the BBF algorithm in the subsequent simulations.

2) Time-Varying Metrics: In this group of simulations, we observe the time-varying metrics of algorithms within the first 200 time slots, while setting the data-receiving capacity of each satellite as 20 Mbits/s. Fig. 6(a)–6(d) demonstrate the time-varying overall amount of data uploaded, energy consumption, penalty and the efficiency of energy consumption, respectively. From Fig. 6(a) and Fig. 6(b), we can see that both the data uploaded and energy consumption increase at the first 20 time slots under all algorithms. However, the proposed online algorithm keeps consistent growing and the two metrics of BBF algorithm with different ζ converge to stable values. The reason is that the limited data-receiving capacity of satellites is completely consumed by uplinks in the first 20 time slots under the BBF algorithm, leading to non-growing data uploaded as well as energy consumption. By contrast, the data-receiving capacity can be well allocated in our proposed algorithm by allocating the corresponding power gears on uplinks, ensuring the data caching in the gateways can keep uploading within the data-receiving capacity 20 Mbits/s in each satellite. As a result, both the overall amount of data uploaded and energy consumption keep growing all the time. However, the penalty of our proposed algorithm decreases as Fig. 6(c) shows. Finally, in Fig. 6(d), we observe that the unit energy consumption under each BBF algorithm has a slow start and a sharp increase afterwards, and finally converges to a stable value. The higher power is consumed while ζ becomes larger, because more energy is needed to reduce more backlogs in queues under BBF algorithm. To the contrast, the proposed online algorithm shows the non-increasing and lowest unit energy consumption for uploading per bit, which implies the highest efficiency of energy consumption.

3) Effect of Data-Receiving Capacity of Satellites: To evaluate the effect of the data-receiving rate capacity of LEO satellites, we set ζ as 10%, and vary the capacity of satellites from 10 to 100 Mbits/s. We then examine the four metrics yielded by algorithms. First, Fig. 7(a) illustrates the data uploaded performance under the proposed online algorithm and the benchmark algorithm at the 100^{th} time slot. It can be seen that the data uploaded shows as a non-decreasing function as the data receiving capacity of satellites grows, and our algorithm outperforms the benchmark algorithm BBF significantly. High data uploaded indicates high total energy consumption, which can be evidenced from Fig. 7(b) when the data receiving capacity is bigger than 30 Mbits/s. When such capacity is lower than 30 Mbits/s under BBF algorithm, to reduce

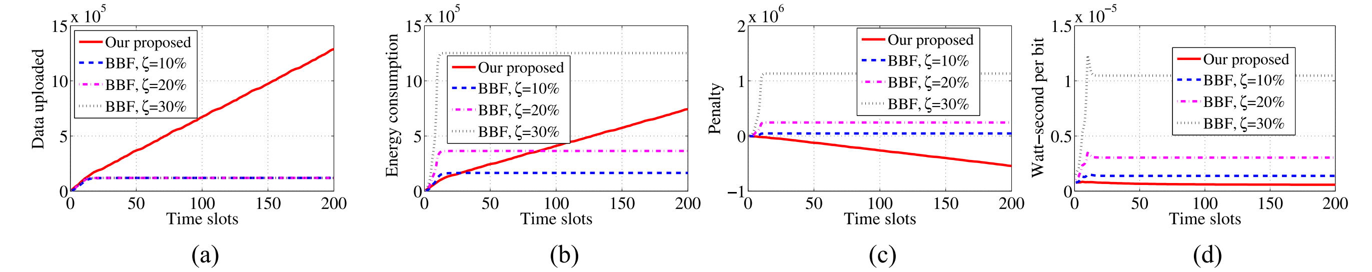


Fig. 6. Comparison of time-varying metrics. Note that, the efficiency of energy consumption reversely associates with the *Watt·second per bit* (*w·s/bit*). The data-receiving capacity of satellites is set to 20 Mbits/s. (a) Data uploaded (Mbits). (b) Energy consumption (*w·s*). (c) Penalty. (d) Watt·second per bit.

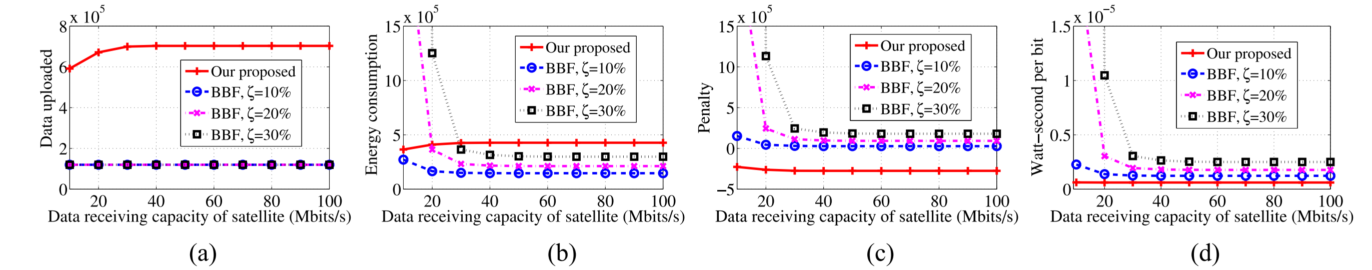


Fig. 7. Performance comparison of online algorithms at the 100^{th} time slot, while varying the data receiving capacity of satellites. (a) Data uploaded (Mbits). (b) Energy consumption (*w·s*). (c) Penalty. (d) Watt·second per bit.

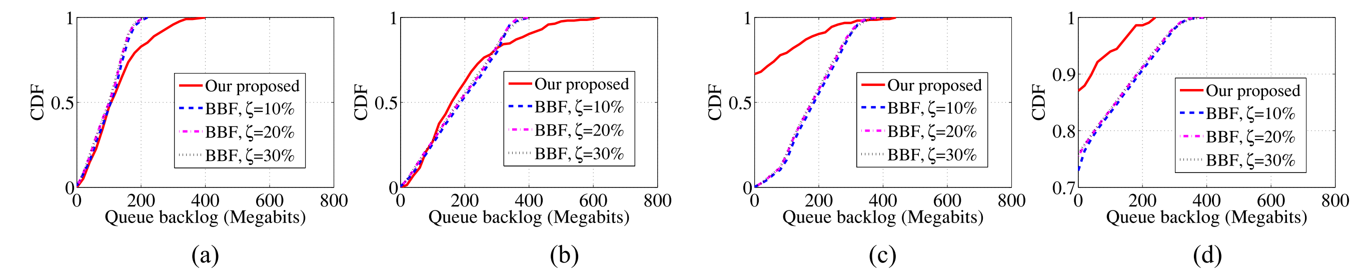


Fig. 8. Cumulative Distribution Function (CDF) of backlogs (shorten as BLs) over all gateway queues at 4 different snapshots. The data-receiving capacity of satellites is set to 10 Mbits/s, $\beta = 1.0$. (a) BLs in the 5^{th} time slot. (b) BLs in the 10^{th} time slot. (c) BLs in the 20^{th} time slot. (d) BLs in the 100^{th} time slot.

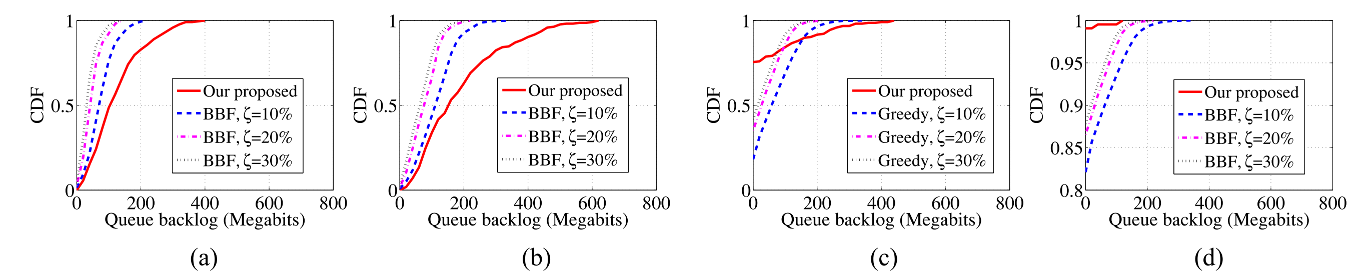


Fig. 9. CDF of backlogs (shorten as BLs) over all gateway queues at 4 different snapshots. The data-receiving capacity of satellites is set to 20 Mbits/s, $\beta = 1.0$. (a) BLs in the 5^{th} time slot. (b) BLs in the 10^{th} time slot. (c) BLs in the 20^{th} time slot. (d) BLs in the 100^{th} time slot.

the specified percentage of backlogs, each gateway needs to exploit many more uplinks than the case when satellite capacity if sufficient enough. As a result, the large number of uplinks consumes high power. This also leads to high penalty, which can be observed in Fig. 7(c). Then, Fig. 7(d) demonstrates the efficiency of energy consumption of algorithms. We can see that the *Watt·second per bit* shows as a non-increasing function of the data-receiving capacity under BBF algorithms. In contrast, the proposed online algorithm achieves a significantly low *Watt·second per bit* measured in 10^{-7}. This implies that our algorithm has a much higher energy efficiency than BBF algorithm does.

4) Backlogs Comparison: By fixing C_j as 10 Mbit/s and 20 Mbit/s, Fig. 8 and Fig. 9 show the Cumulative Distribution Functions (CDFs) of the queue backlogs over all gateways at four moments, respectively. We observe some interesting findings in both the two figures. First, the number of large queue backlogs under all algorithms increases as time goes

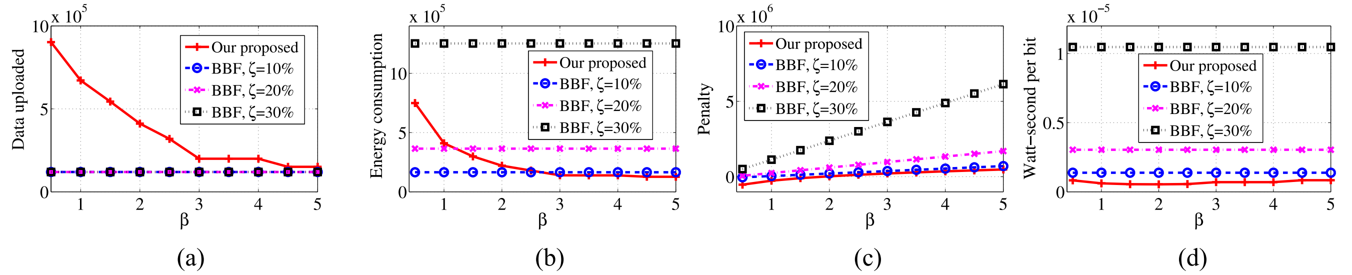


Fig. 10. Performance comparison of online algorithms at the 100^{th} time slot, while varying the weight of energy consumption, i.e., β. The data-receiving capacity of satellites is set to 20 Mbits/s. (a) Data uploaded (Mbits). (b) Energy consumption ($w{\cdot}s$). (c) Penalty. (d) Watt·second per bit.

by in the first 10 time slots. This is due to the fact that a small part of IoT gateways are out of reach to any satellites in the first few time slots. The IoT data stream keeps arriving in those part of gateways, making their backlogs grow. However, the number of queues with empty-backlog increases drastically when system operates under our proposed algorithm after the 10^{th} time slot. The reason is that the number of available time-varying uplinks increases gradually. Thus, the arrival IoT data caching in the queues of the connected gateways can be uploaded quickly. In contrast, the backlogs in all queues keep growing under the benchmark BBF algorithm. Because only very few portion of gateways can upload their data via the uplinks. This leads to that the other gateways have an ever-growing backlogs.

On the other hand, through the comparison between Fig. 8 and Fig. 9, we also find that the backlogs tend to be smaller when the data receiving capacity of satellites is changed from 10 Mbits/s to 20 Mbits/s.

5) Effect of β: We finally evaluate the effect of β, i.e., the weight of energy consumption in the penalty function. Fig. 10 shows the four metrics at the 100^{th} time slot, while varying β from 0.5 to 5. It is shown that only the penalty grows while β increases under BBF algorithm. The large ζ indicates high energy consumption and low efficiency of energy usage. In contrast, in the proposed algorithm, both overall amount of data uploaded and total energy consumption exhibit as non-increasing function as β grows. Because the weight of energy consumption becomes big in the penalty function when enlarging β. Thus, the energy consumption is reduced, while the penalty is increased. Interestingly, we also find that the performance of *Watt·second per bit*under the proposed algorithm achieves the lowest point when β is equal to 2.5 as shown in Fig. 10(d). This implies that $\beta = 2.5$ is the most efficient choice for the current system settings in terms of energy consumption.

In summary, the proposed online scheduling algorithm achieves larger overall amount of data uploaded and higher efficiency of energy-consumption, and also yields significant smaller queue backlogs than those of the benchmark BBF algorithm.

VI. Conclusion

In this paper, we studied how to gather IoT data from geo-distributed networks in an energy-efficient way, based on the LEO based communication networks. We proposed an online scheduling algorithm to address this challenge. The extensive simulation results show that the proposed algorithm can achieve much higher efficiency of energy consumption while maintaining significant lower queue backlogs in IoT gateways, compared with a greedy "Big-Backlog-First" heuristic algorithm.

References

[1] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2016, pp. 1–5.

[2] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 66–74, Apr. 2011.

[3] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[4] H. S. Dhillon, H. Huang, and H. Viswanathan, "Wide-area wireless communication challenges for the Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 168–174, Feb. 2017.

[5] R. Barbatei, A. Skavhaug, and T. A. Johansen, "Acquisition and relaying of data from a floating wireless sensor node using an unmanned aerial vehicle," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Denver, CO, USA, 2015, pp. 677–686.

[6] A. Zolich, J. A. Alfredsen, T. A. Johansen, and K. R. Skøien, "A communication bridge between underwater sensors and unmanned vehicles using a surface wireless sensor network-design and validation," in *Proc. OCEANS Shanghai*, Shanghai, China, 2016, pp. 1–9.

[7] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of Remote Things," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[8] (2018). *IsatData Pro*. [Online]. Available: https://www.inmarsat.com/service/isatdata-pro/

[9] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[10] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the Internet of Things," in *Proc. IEEE 14th Int. Symp. Workshops World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2013, pp. 1–6.

[11] X. Luo, G. Liu, and H. Huang, "Buffer capacity-constrained epidemic routing model in mobile ad-hoc networks," in *Proc. IEEE 17th Int. Conf. Comput. Sci. Eng. (CSE)*, 2014, pp. 1443–1448.

[12] P. Gonizzi, G. Ferrari, V. Gay, and J. Leguay, "Data dissemination scheme for distributed storage for IoT observation systems at large scale," *Inf. Fusion*, vol. 22, pp. 16–25, Mar. 2015.

[13] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems* (Synthesis Lectures on Communication Networks), vol. 3. San Rafael, CA, USA: Morgan & Claypool, 2010, pp. 1–211.

[14] C.-T. Cheng, N. Ganganath, and K.-Y. Fok, "Concurrent data collection trees for IoT applications," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 793–799, Apr. 2017.

[15] H. Wu, J. Li, H. Lu, and P. Hong, "A two-layer caching model for content delivery services in satellite-terrestrial networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[16] X. Jia, T. Lv, F. He, and H. Huang, "Collaborative data downloading by using inter-satellite links in LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1523–1532, Mar. 2017.

[17] M. Cello, M. Marchese, and F. Patrone, "Coldsel: A selection algorithm to mitigate congestion situations over nanosatellite networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[18] H. Huang, S. Guo, W. Liang, and K. Wang, "Online green data gathering from Geo-distributed IoT networks via leo satellites," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[19] H. Huang, S. Guo, W. Liang, K. Wang, and Y. Okabe, "Coflow-like online data acquisition from low-earth-orbit datacenters," *IEEE Trans. Mobile Commun.*, to be published.

[20] D. Xu and X. Liu, "Geographic trough filling for Internet datacenters," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2012, pp. 2881–2885.

[21] J. P. Choi and V. W. S. Chan, "Predicting and adapting satellite channels with weather-induced impairments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 779–790, Jul. 2002.

[22] Z. Na, Q. Guan, C. Fu, Y. Cui, and Q. Guo, "Channel model and throughput analysis for LEO OFDM satellite communication system," *Int. J. Fut. Gener. Commun. Netw.*, vol. 6, no. 6, pp. 109–122, 2013.

[23] A. Guidotti *et al.*, "LTE-based satellite communications in LEO mega-constellations," *Int. J. Satellite Commun. Netw.*, pp. 1–15, Jul. 2018. [Online]. Available: https://doi.org/10.1002/sat.1258

[24] L. J. Ippolito and L. J. Ippolito, Jr., *Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance*. Chichester, U.K.: Wiley, 2017.

[25] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power allocation and routing in multibeam satellites with time-varying channels," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

[26] (2019). *Technical Report*. [Online]. Available: https://drive.google.com/file/d/1wWKm0MP7v_FK8RG0Xy89a1lnfu6t6iaP/view?usp=sharing

[27] (2019). *Systems Tool Kit*. [Online]. Available: https://en.wikipedia.org/wiki/Systems_Tool _Kit/

[28] J. Wang, L. Li, and M. Zhou, "Topological dynamics characterization for LEO satellite networks," *Comput. Netw.*, vol. 51, no. 1, pp. 43–53, 2007.

Huawei Huang (M'16) received the Ph.D. degree in computer science and engineering from the University of Aizu, Japan. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, China. He was a Visiting Scholar with Hong Kong Polytechnic University from 2017 to 2018, a Post-Doctoral Research Fellow of JSPS from 2016 to 2018, and an Assistant Professor with Kyoto University, Japan, from 2018 to 2019. His research interests mainly include SDN/NFV and edge computing. He was a recipient of the Best Paper Award in IEEE TrustCom-2016.

Song Guo (M'02–SM'11) received the Ph.D. degree in computer science from the University of Ottawa. He is currently a Full Professor with the Department of Computing, Hong Kong Polytechnic University. His research has been sponsored by JSPS, JST, MIC, NSF, NSFC, and industrial companies. His research interests are mainly in the areas of cloud and green computing, big data, wireless networks, and cyber-physical systems. He has published over 300 conference and journal papers in the above areas. He was a recipient of multiple best paper awards from IEEE/ACM conferences. He has served as an Editor of several journals, including the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, the *IEEE Communications Magazine*, and *Wireless Networks*. He has been actively participating in international conferences serving as the general chair and the TPC chair. He is a Senior Member of ACM.

Weifa Liang (M'99–SM'01) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China in 1989, and the Ph.D. degree in computer science from Australian National University in 1998, where he is currently a Professor with the Research School of Computer Science. His research interests include design and analysis of energy-efficient routing protocols for wireless ad hoc and sensor networks, cloud computing, software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He is a member of ACM.

Kun Wang (M'13–SM'17) received the first Ph.D. degree in computer science from the Nanjing University of Posts and Telecommunications, China, in 2009 and the second Ph.D. degree in computer science from the University of Aizu, Japan, in 2018. He was a Post-Doctoral Fellow with the University of California at Los Angeles (UCLA), USA, from 2013 to 2015, and a Research Fellow with Hong Kong Polytechnic University, Hong Kong, from 2017 to 2018. He is currently a Research Fellow with UCLA. His current research interests are mainly in the area of big data, wireless communications and networking, energy Internet, and information security technologies. He was a recipient of the IEEE GLOBECOM 2016 Best Paper Award, the IEEE TCGCC Best Magazine Paper Award in 2018, and the IEEE ISJ Best Paper Award in 2019. He serves as an Associate Editor for IEEE ACCESS, an Editor for the *Journal of Network and Computer Applications*, and a Guest Editor for IEEE NETWORK, IEEE ACCESS, *Future Generation Computer Systems*, *Peer-to-Peer Networking and Applications*, *IEICE Transactions on Communications*, the *Journal of Internet Technology*, and *Future Internet*.

Albert Y. Zomaya (M'90–SM'97–F'04) is currently the Chair Professor of the High Performance Computing and Networking, School of Computer Science, University of Sydney, where he is the Director of the Centre for Distributed and High Performance Computing. He published over 600 scientific papers and articles. He has authored, coauthored, or edited over 20 books. His research interests are in the areas of parallel and distributed computing, networking, and complex systems. He was a recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award in 2011, the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing in 2011, and the IEEE Computer Society Technical Achievement Award in 2014. He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON COMPUTERS from 2011 to 2014. He is the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. He currently serves as an Associate Editor for 22 leading journals, such as *ACM Computing Surveys*, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the *Journal of Parallel and Distributed Computing*. He is a fellow of AAAS and IET, U.K.