Achieving Ultra Energy-efficient and Collision-free Data Collection in Wake-up Radio Enabled mIoT

Chia-An Hsu*, Frank Y. Li§, Chiuyuan Chen*, and Yu-Chee Tseng‡
*Department of Applied Mathematics, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
§Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
‡Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
Email: ab89414@gmail.com; frank.li@uia.no; cychen@mail.nctu.edu.tw; yuchee@cs.nctu.edu.tw

***Abstract*—To achieve ultra-low energy consumption and decade-long battery lifetime for Internet of things (IoT) networks, wake-up radio (WuR) appears as an eminent solution. While keeping devices in deep sleep for most of the time, a WuR enabled IoT device can be woken up for data transmission at any time by a wake-up call (WuC). However, collisions happen among WuCs for *transmitter-initiated data reporting* and among data packets for *receiver-initiated data collection*. In this paper, we propose two novel hashing-based schemes in order to achieve collision-free data transmissions for receiver-initiated data collection. We consider first a simple scenario in which all devices in a region of interest are covered by a data collector and propose a scheme which facilitates a scheduled time for data uploading of each device. Then we extend our scheme to cover a more realistic scenario where IoT devices are distributed across a larger region that cannot be covered by a single data collector. In this case, we propose a partitioning algorithm for data collection across multiple partitions. Both analysis and simulations are performed to demonstrate the effectiveness of the proposed schemes.**

I. Introduction

To connect a huge number of Internet of things (IoT) devices, known as massive IoT (mIoT), two categories of connections are most appealing, i.e., cellular network based connections such as long term evolution for machine-type communication (LTE-M) and narrowband IoT (NB-IoT); and non-cellular based connections such as long range wide area network (LoRaWAN). No matter which solution is adopted, energy consumption always plays a pivotal role for IoT network design since these devices are typically battery powered and are often deployed at places that are not easily accessible.

Traditionally, data collection in IoT networks adopts duty-cycling (DC) mechanisms which can significantly save energy consumption, however, at a cost of long delay. Recently, a paradigm shift from DC medium access control (MAC) to wake-up radio (WuR) is undergoing thanks to the overwhelming superiority for energy consumption provided by WuR [1]-[3]. Different from the DC principle, the novelty of WuR operation is to associate the main radio (MR) of an IoT device with a wake-up receiver (WuRx) which consumes power at the magnitude of 1000 times less than that of the MR. While the MR is in deep sleep for most of the time and wakes up for data transmission in an on-demand manner, the WuRx is always on listening to any activities over the channel.

To initiate data transmission in WuR enabled IoT networks, there are two working modes, i.e., transmitter-initiated (TI) *data reporting* and receiver-initiated (RI) *data collection*. In both cases, the activity of the MR of a device will be triggered by a wake-up call (WuC) received by its associated WuRx. A WuC is a specifically designed message dedicated to wake a target device up for receiving or transmitting a data frame when an event occurs. When multiple devices initiate their WuCs at the same time (in the TI mode) or a broadcast/multicast WuC is received by multiple devices (in the RI mode), collision(s) may happen.

Intensive studies on data transmission and protocol design in WuR enabled non-cellular IoT networks have been performed recently [4][5]. In [6] and [7], four schemes for WuC collision avoidance have been proposed for TI or RI based networks respectively. Through clear channel assessment (CCA) and backoff, much lower collision probabilities have been achieved. However, contention among IoT devices still exists during the transmission procedure, especially when the number of devices is large. To avoid collision for data collection (or uploading) in RI based networks, which is the focus of this study, one may consider to use unicast WuC to poll each IoT device consecutively. However, such a procedure leads to long delay since WuCs are transmitted at a much lower data rate [1] [3]. On the other hand, a contention-based uploading mechanism may also cause long delay due to backoff and comparatively high energy consumption due to CCA.

In this paper, we propose two energy-efficient and collision-free data collection schemes for non-cellular WuR-enabled IoT networks in which a data collector could be a fixed station or an unmanned aerial vehicle (UAV). For each round of data collection, a dedicated WuC is multicast by the UAV to a group of IoT devices. The specially designed WuC contains the information for multiple devices through which the time slot assigned to each device being involved can be deduced. To realize this idea, we leverage a hash function through which each device can derive its dedicated time slot for transmission. Furthermore, we extend our scenario to a large-scale network in which the coverage of the UAV cannot cover all devices. In this case, the data collector categorizes the devices into multiple subgroups. Then we propose a spanning forest based heuristic algorithm which is able to obtain an optimal partition for devices. Via both analysis and simulations, we demonstrate that our schemes can effectively avoid collisions and shorten transmission time, leading to a longer lifetime for devices.

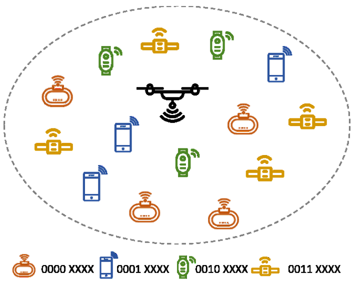


Fig. 1: Scenario 1: All devices are covered by a UAV but multiple groups exist. 4 groups, each with its pre-configured netmask, are illustrated as Red($\mathcal{S}_1$), Blue($\mathcal{S}_2$), Green($\mathcal{S}_3$), and Yellow($\mathcal{S}_4$) respectively.

The rest of this paper is organized as follows. After explaining the network scenarios in Sec. II, we present the proposed schemes in Sec. III. The performance of the schemes is analyzed in Sec. IV, followed by simulation results presented in Sec. V. Finally, the paper is concluded in Sec. VI.

II. Network Scenarios and Problem Statement

Two network scenarios for RI data collection are evisaged, as illustrated in Fig. 1 and Fig. 2 respectively.

Scenario 1: Consider a set of devices, denoted by $\mathcal{S}$, that can be covered by one UAV, e.g., when it hovers for a moment to collect data from a garden. The type of data to be collected is heterogeneous. Therefore, the devices are categorized into multiple groups by their functions (e.g., weather monitoring and asset tracking devices belong to two different groups) such that $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$, where each S_i is a group and $S_i \cap S_j = \emptyset$ if $i \neq j$. Furthermore the addresses of the devices in the same group are configured under one common subnet mask.

When a WuC is sent from the UAV, all devices will receive this WuC. Accordingly, an energy-efficient data collection procedure needs to be designed to wake the target devices up and for uploading their data such that the total energy consumption will be minimized. The total energy consumption is defined as the sum of the energy consumption by all devices.

Scenario 2: Consider a more complicated scenario where the deployment region is much larger than the coverage of a WuC. In this case, the UAV may not be able to collect data from the same group of devices by sending just one WuC.

Accordingly, the UAV has to initiate multiple WuCs for data collections, each for an area covered by one WuC. Accordingly, a partitioning algorithm needs to be designed in order to find the optimal partition of the region that minimizes total energy consumption. For data collection in each partition, the same procedure for Scenario 1 applies.

III. Proposed RI based Data Collection Schemes

The main idea of our schemes is to schedule the transmissions of IoT devices to different time instants through a common WuC sent by the data collector. To do so, each device obtains its time slot by computing a simple hash function based on a common seed multicast by the data collector via a WuC. Correspondingly, the collector avoids collisions among transmissions of end devices at a cost of slightly higher overhead when computing the seed.

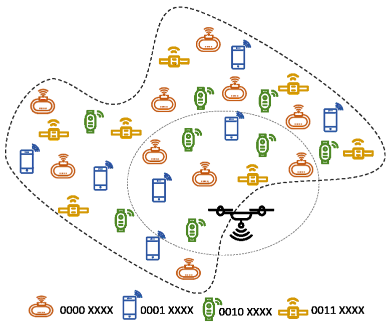


Fig. 2: Scenario 2: Devices are distributed across a much larger region and network partition is expected for data collection.

A. Proposed Scheme 1

Considering Scenario 1 operated under the RI mode, Fig. 3 illustrates that a mission consisting of n rounds of data collections initiated by a UAV. One round, or a cycle, represents the data collection procedure initiated through one WuC.

Within one cycle we divide the whole duration into multiple time slots. The duration of the cycle as well as the length of each time slot are configurable. As illustrated in Fig. 4, each cycle is composed of two frames: a *scheduled frame* (SF) and a *random frame* (RF). The numbers of time slots needed in each SF and RF, denoted by L and M respectively, depend on device population and these numbers are pre-configured. The length of each time slot is the sum of time needed for transmitting a data frame and an ACK or a NACK frame.

1) Protocol Design: In Scenario 1, the devices are classified into mutually exclusive groups with different subnet masks. For data transmission in a specific group, each time slot in the SF is uniquely assigned to one device by the hash function so that it can upload its data without contention.

To facilitate collision-free transmissions, we embed a common hash function among the UAV and devices. The hash function takes an ID, a common seed, and a frame length of L as the input and computes an integer value between $[1, L]$. To perform the hash function, each device uses its own ID as an input. The seed is decided and sent by the collector via a WuC. An example of such an operation is shown in Fig. 5.

However, finding an ideal seed for the hash function, i.e., a seed that does not produce any collision, is not an easy process [8], especially when the device population is large. To address this potential problem, we introduce another frame,

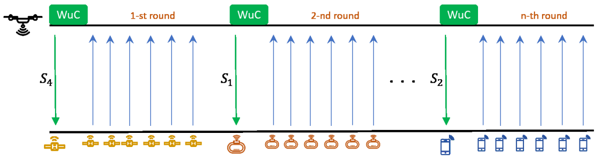


Fig. 3: A mission with n rounds of data collection.

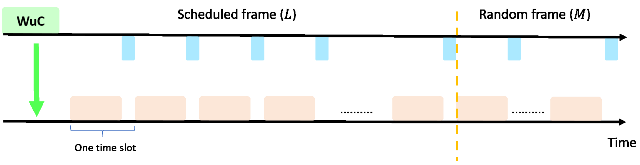


Fig. 4: Two frames exist in one round of data collection.

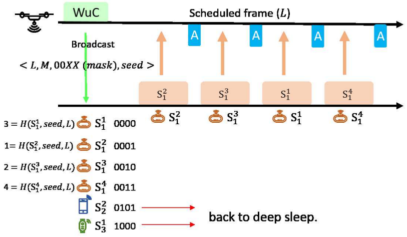

WuC
Scheduled frame (L)
Broadcast
$< L, M, 00XX\,(mask), seed >$
A
A
A
A
S_1^2
S_1^3
S_1^1
S_1^4
S_1^2
S_1^3
S_1^1
S_1^4
$3 = H(S_1^1, seed, L)$ S_1^1 0000
$1 = H(S_1^2, seed, L)$ S_1^2 0001
$2 = H(S_1^3, seed, L)$ S_1^3 0010
$4 = H(S_1^4, seed, L)$ S_1^4 0011
S_2^2 0101
S_3^1 1000
back to deep sleep.

Fig. 5: Use hashing to assign time slot to each device.

known as RF, and append it immediately after the SF. The purpose of introducing an RF is to resolve a collision that occurred during the SF through random access.

To collect data from a specific group, $\mathcal{S}_i$, the proposed protocol works as presented below.

- Step 1: Prior to a data collection mission, the collector selects a seed for the hash function and decides the frame lengths of L and M, as well as the mask of $\mathcal{S}_i$. It then initiates a cycle by broadcasting a WuC to all devices.
- Step 2: Upon receiving the broadcast WuC, each device in $\mathcal{S}$ wakes from the *deep sleep mode* to the *light sleep mode* and checks whether it belongs to this group by comparing the mask. Since each group is covered by a distinct mask, only the devices in $\mathcal{S}_i$ will go to *Step 3*. Other devices will go back to the *deep sleep mode*. Furthermore, since the length of a netmask is much shorter than the length of a full address, those devices that are not in $\mathcal{S}_i$ will terminate decoding and go back to the deep sleep mode in a much early stage.
- Step 3: The devices in $\mathcal{S}_i$ then compute the hash function $\mathcal{H}(ID_k, seed, L)$ based on its own ID, L, and the seed. The retrieved value from $\mathcal{H}(ID_k, seed, L)$ is the index of a time slot that is assigned to this device in the SF which contains the first L time slots in the frame structure.
- Step 4: After computing the time slot index, the device stays in the *light sleep mode* until it transmits at time slot $\mathcal{H}(ID_k, seed, L)$. Then it waits for an ACK.
- Step 5: If an ACK is received, the device goes back to the *deep sleep mode*. Otherwise, it stays in the *light sleep mode*. Note that a NACK might be caused by a hashing collision or erroneous decoding.
- Step 6: The devices which failed in the scheduled frame will randomly select a time slot in the subsequent RF to upload their data again. Then, it goes back to deep sleep.

In Fig. 6, we give a simple example to illustrate the operation by considering a group of devices, S_1, that has a netmask as 00. Among the six devices shown in the figure, two of them which have a netmask as 01 or 10 respectively will go to deep sleep right after decoding the netmak. The other four devices will continue to retrieve the time slot allocated to them. A collision happens since two devices retrieve a common time slot in the scheduled frame, but the collision is resolved by two separate transmissions in the subsequent RF.

2) Parameter Configuration: For any given number of N devices, the lengths of the SF and RF, i.e., L and M, need to be configured properly. With respect to L, a higher success probability in the SF will be achieved when L increases. However, a longer frame leads to higher power consumption and longer transmission time. Herein, we define a configurable parameter λ such that $L = \lambda N$. For a specific network, an appropriate value for λ needs to be configured.

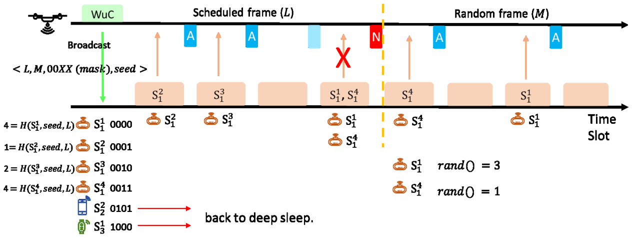


Fig. 6: The proposed wake-up protocol for RI data collection.

To decide a proper value for M, it is preferable to configure it to a comparatively short length since the purpose of an RF is to provide a second chance to those devices which failed in the SF. To do so, we identify an optimal value for M, denoted as M^*, which optimizes the time slot utilization in the RF. The definition of time slot utilization $U(M, N, \alpha)$ and the procedure of finding M^* are explained as follows (since we are determining M for a given N and α, the only variable in function $U(M, N, \alpha)$, denoted hereafter as $U(M)$, is M):

$$\begin{aligned} U(M) &\triangleq \frac{E[\text{number of success in an RF}]}{\text{number of allocated time slots in the RF}} \\ &= \frac{E[\text{no. of devices in the RF}]\ P(\text{success in the RF})}{M} \\ &= \frac{N\alpha\ (1-1/M)^{N\alpha-1}}{M}, \end{aligned} \tag{1}$$

where α is the hash function collision probability and $0 < \alpha < 1$. To configure M optimally, we need to find

$$M^* \triangleq \underset{M>0}{\arg\max}\ U(M). \tag{2}$$

To find a global optimal value for M, we take the first order derivative of $U(M)$ w.r.t. M and obtain

$$\frac{\mathrm{d}U(M)}{\mathrm{d}M} = \frac{N\alpha\,(1-1/M)^{N\alpha-2}}{M^3}\,(N\alpha - M)\,. \tag{3}$$

From (3), it is clear that $M = N\alpha$ is a critical point since $\mathrm{d}U(M)/\mathrm{d}M$ has a positive or negative value when $M < N\alpha$ or $M > N\alpha$ respectively. Thus, we can conclude that $M^* = N\alpha$ is a global optimal value.

B. Proposed Scheme 2

Based on Scheme 1 presented above, we propose another scheme targeting at Scenario 2. In order to perform RI based data collection in this case, the data collector needs to divide the region into multiple partitions. For each partition, the data collection procedure is the same as presented in Scheme 1 since multiple groups of devices may exist in each partition.

Accordingly, multiple missions are required in order to collect data across the whole region. Therefore, how to divide the region or device groups into multiple partitions is essential for Scheme 2 since different partition leads to different level of energy consumption. In what follows, we propose an algorithm which finds an optimal partition for a given network.

The thrust of this algorithm is to divide $\mathcal{S}_i$ by device locations through an enhanced Prim's algorithm [9]. The original algorithm forms a minimum spanning tree for a *connected* weighted graph by adding a closest vertex to the current subtree at each step. In this work, we modify the algorithm by checking whether the diameter of the current subtree at each step exceeds the coverage of the data collector after adding any vertex. If yes, i.e., the current subtree cannot be further extended, the established subtree is stored and the algorithm starts from a new vertex that has not been added into any subtree. The enhanced algorithm retrieves a spanning forest in which each subtree represents one partition of the original device group. Different from the original Prim's algorithm which starts from an arbitrary vertex, our algorithm starts from a vertex that is closest to the previously selected subgroup S_{i_k} to form a new subtree. This property can guarantee that the new subtree starts with an optimal choice. Furthermore, note that the "if" condition in Line 10 of Algorithm 1 can be checked by using the following theorem [10].

Theorem 1. *Each set $\mathcal{X} \subset \mathbb{R}^2$ of diameter at most d (i.e., any two points have a distance at most d) is contained in some disc of radius $d/\sqrt{3}$.*

Consider now a moving data collector, e.g., a UAV, and a set of devices $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$. For a given group $\mathcal{S}_i$, our aim is to find the optimum partition for this group and the corresponding location for WuC broadcasting for each partition set $\mathcal{S}_{i_j}, j = 1, ..., k$, such that the total energy consumption of these devices is minimized. The problem can be formulated as an optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{D}} \quad & Energy(\mathcal{S}_i) = \sum_{s \in \mathcal{S}_i} Energy(s, \boldsymbol{D}) \\ \text{s.t.} \ & \forall j \in \{1, ..., k\}, \text{ all devices} \in \mathcal{S}_{i_j} \text{ can be} \\ & \quad \text{covered by a WuC sent at } (x_j, y_j) \in \mathbb{R}^2. \end{aligned} \tag{4}$$

Here $\boldsymbol{D} = (P, L)$ is a vector that consists of the partition result $P = (\mathcal{S}_{i_1}, ..., \mathcal{S}_{i_k})$ and the corresponding WuC locations $L = ((x_1, y_1), ..., (x_k, y_k))$. Note that $\mathcal{S}_{i_h} \cap \mathcal{S}_{i_l} = \emptyset$ for $h \neq l$ and $\bigcup_{j=1}^{k} \mathcal{S}_{i_j} = \mathcal{S}_i$.

After solving the optimization problem, we obtain a solution $\boldsymbol{D}^* = (P^*, L^*)$ which represents the optimal partition. Hereafter, Scenario 2 can be represented by a collection of multiple groups under Scenario 1. In other words, the rest of Scheme 2 works the same as or similar to Scheme 1. However, how to design a travel route for flying UAV across the region of interest is beyond the scope of this paper.

IV. Performance Analysis

In this section, we analyze the performance of our schemes with respect to three parameters, i.e., access success probability, access delay, and energy consumption. Closed-form expressions are deduced based on our definitions.

Algorithm 1: Partition Algorithm

```
   Input: $\mathcal{S}_i$ and the location of each device in $\mathcal{S}_i$
   Output: $\boldsymbol{D} = (P, L)$
1  Construct a complete weighted graph $G = (V, E)$ by input
       $\mathcal{S}_i$ /* where $V = \mathcal{S}_i$ and each $e \in E$ represents
       the distance between two devices in $\mathcal{S}_i$.    */
2  $V_R \leftarrow V$; $E_R \leftarrow E$; $k \leftarrow 0$.
3  while $V_R \neq \emptyset$ do
4      Find $u \in V_R$ that is closest to $S_{i_k}$. /* choose any
           $u \in V_R$ if $k = 0$.                              */
5      $k \leftarrow k + 1$.
6      $S_{i_k} \leftarrow \{u\}$.
7      $E_{cut} \leftarrow \{e = xy \in E_R \,|\, x \in S_{i_k}, y \in V_R \setminus S_{i_k}\}$.
8      while $E_{cut} \neq \emptyset$ do
9          Find min $e_m = xy \in E_{cut}$.
10         if $S_{i_k} \cup \{y\}$ has diameter $\leq$ (UAV radius)$\sqrt{3}$ then
11             $S_{i_k} \leftarrow S_{i_k} \cup \{y\}$.
12             $E_{cut} \leftarrow E_{cut} \setminus \{e = ya \,|a \in S_{i_k}\}$.
13             $E_{cut} \leftarrow E_{cut} \cup \{e = yb \,|b \in V_R \setminus S_{i_k}\}$.
14         else
15             $E_{cut} \leftarrow E_{cut} \setminus \{e_m\}$.
16     $(x_k, y_k)$ = the middle point of $S_{i_k}$.
17     $V_R \leftarrow V_R \setminus S_{i_k}$.
18     $E_R \leftarrow E_R \setminus \{e \,|\, e$ has endpoint in $S_{i_k}\}$.
19 $L \leftarrow ((x_1, y_1), ..., (x_k, y_k))$; $P \leftarrow (S_{i_1}, ..., S_{i_k})$.
20 return $\boldsymbol{D} = (P, L)$
```

A. Access Success Probability

Consider a randomly selected device as a reference device (RD). We calculate the access success probability for this device in one round, denoted by P_{rd}, as follows.

For the success access of the RD, its successful transmission could happen during either the SF or the RF. Therefore, P_{rd} is equal to the sum of success probability in the SF and the success probability in the RF. Recalling that α is the hash function collision probability, P_{rd} can be expressed as follows.

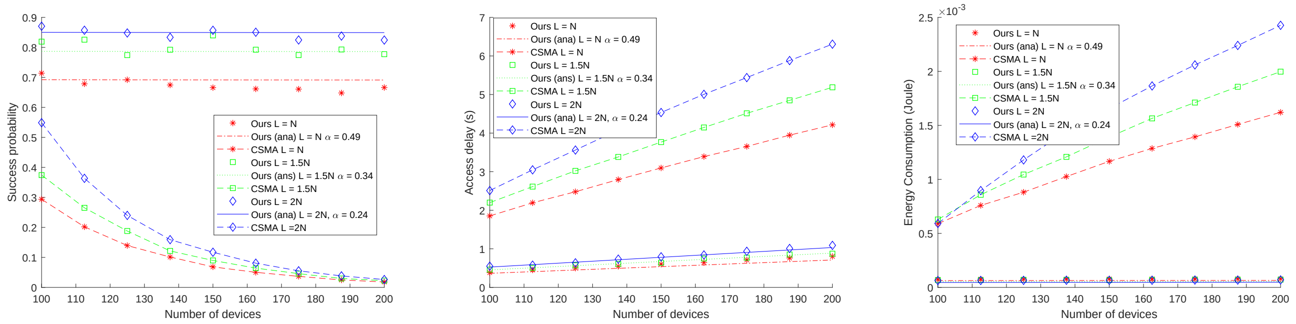
$$P_{rd} = P_s^{sf} + (1 - P_s^{sf})P_s^{rf} = (1 - \alpha) + \alpha(1 - \frac{1}{M})^{N\alpha} - 1 \tag{5}$$

where P_s^{sf} and P_s^{rf} stand for the successful transmission probability during the SF and RF, respectively.

B. Access Delay

The access delay of the RD, T_{rd}, is defined as the expected duration from WuC broadcasting to the successful transmission of the RD. Denote by T_s the length of a time slot. Note that both data frame and ACK/NACK transmissions happen within the same time slot. Let P_i^{sf} and P_j^{rf} be the successful transmission probability during the i-th slot of the SF and the j-th slot of the RF respectively. T_{rd} can be obtained as:

$$\begin{aligned} T_{rd} &= \sum_{i=1}^{L} i T_s P_i^{sf} + \sum_{j=1}^{M} (j + L) T_s P_j^{rf} \\ &= T_s \left(\frac{(1 - \alpha)(L + 1)}{2} + \frac{\alpha(1 + 2L + M)}{2} \right) + T_{wuc} \end{aligned} \tag{6}$$



(a) Scheme1: Access success probability.

(b) Scheme 1: Access delay.

(c) Scheme 1: Energy consumption.

Fig. 7: Performance of Scheme 1 in term of access success probability, access delay, and energy consumption as N varies.

where T_{wuc} denotes the WuC decoding duration.

C. Energy Consumption

Consider that the RD belongs to the target group for a cycle of data transmission. After the reception of a WuC which consumes E_{wuc} amount of energy, the RD needs to stay in the light sleep mode until its scheduled time slot arrives. Assuming that it transmits during the i-th slot of the SF, it will then consume energy in the light sleep mode for $i-1$ slots and in the active mode for one slot. If there is a hash function collision, the transmission will continue in the RF. This means that the RD will stay in the active mode for 2 time slots in total. Therefore, the expected energy consumption of the RD during one cycle, E_{rd}, is obtained as follows

$$\begin{aligned}
E_{rd} &= T_s \sum_{i=1}^{L}((i-1)P_{ls} + P_a)P_i^{sf} \\
&+ T_s \sum_{j=1}^{M}((j+L-2)P_{ls} + 2P_a)P_j^{rf} \\
&= (1-\alpha)\left(\frac{P_a + (L-1)\,P_{ls}}{2}\right)T_s \\
&+ \alpha\left(\frac{P_{ls}\,(M+2L-3) + 4P_a}{2}\right)T_s + E_{wuc} \qquad (7)
\end{aligned}$$

where P_{ls} and P_a denote the power consumption levels for light sleep and active respectively.

V. Simulations and Numerical Results

To evaluate the performance of the proposed schemes, we implement the schemes in MATLAB and conduct extensive simulations. The performance of our schemes is compared with that of the carrier sense multiple access with collision avoidance (CSMA/CA)-based access (for Scheme 1) and another reference scheme known as *Square* (for Scheme 2) with respect to the three parameters defined above. For network configuration, we vary the number of devices in the region, N, from 100 to 200. For each frame transmission, a packet size of 1000 bits transmitted at the data rate of 250 kbps is configured. Furthermore, three values of L and α are configured. Based on the value of N, the optimal value of M is decided by $M^* = N\alpha$. To evaluate the effect of frame length in our scheme, we configure λ as $\lambda = 1$, 1.5, and 2 respectively. From the curves shown in the numerical result figures, it is evident that the simulation results match the analytical ones closely. Thus the correctness of our analysis is confirmed.

A. Performance of Scheme 1 as Device Population Varies

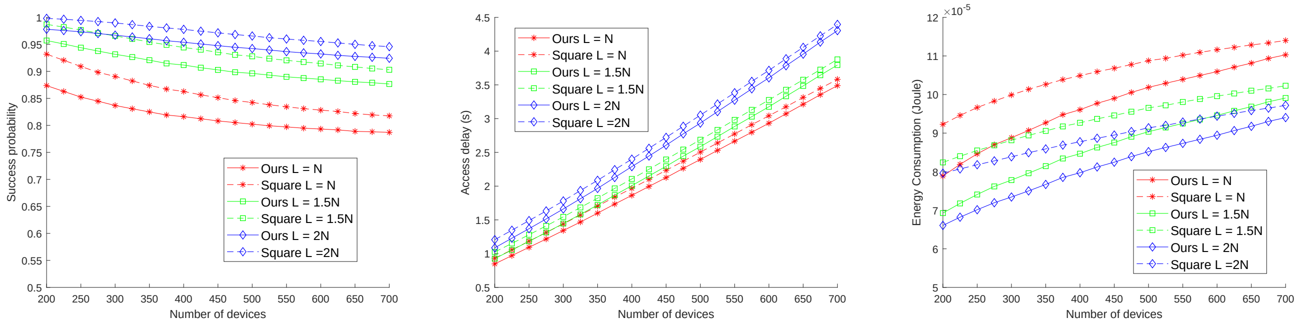
1) Access Success Probability: To make a fair comparison, we configure the number of time slots for CSMA/CA the same as what is used in our scheme. The obtained results for access success probability as the number of devices varies are shown in Fig. 7a).

As shown in the figure, the access success probabilities obtained by our scheme remain stable as the number of devices increases whereas the CSMA/CA-based probabilities decrease monotonically. This is because the employment of a hash function assigns dedicated time slots for data uploading and can effectively avoid collisions. In contrast, the CSMA/CA-based access scheme relies solely on random access causing lots of collisions and the collision problem becomes worse as the device population grows.

2) Access Delay: According to the design of our scheme, a longer frame length is needed with a larger number of devices. Correspondingly, a longer delay is expected as the number of devices N increases. However, the slope of the delay increase in our scheme is not as sharp as in CSMA/CA and the obtained delay from our scheme is shorter for the whole range of device population, as illustrated in Fig. 7b).

The reason for this behavior is as follows. To resolve collisions in CSMA/CA, multiple retransmissions of the same frame are allowed. The larger the device population, the higher the number of retransmissions. Consequently, much longer access delay is experienced due to retransmissions. With our scheme, a longer frame length is allocated based on the value of α. Since our scheme requires typically only one transmission, the achieved delay is still much shorter than the one obtained in CSMA/CA.

Furthermore, it is worth mentioning that the achieved delay in our scheme is also less than the traditional unicast based data collection for WuR enabled IoT networks. This is because WuCs are transmitted at a much lower data rate [3]. Instead of sending a WuC individually to each device, our schemes require only one WuC for a group of devices.



(a) Scheme 2: Access success probability.

(b) Scheme 2: Access delay.

(c) Scheme 2: Energy consumption.

Fig. 8: Performance of Scheme 2 in term of access success probability, access delay, and energy consumption as N varies.

3) Energy Consumption: The energy consumption performance of our scheme is compared with that of CSMA/CA in Fig. 7c). As expected, our scheme outperforms CSMA/CA overwhelmingly for the whole range of device population. The reason is as follows.

For each frame transmission in CSMA/CA, a device has to sense the channel and perform backoff before each data frame transmission. The channel sensing and backoff procedures also consume a certain amount of energy at the mW power level. On the other hand, our scheme requires WuC decoding and devices wait for scheduled transmission only at the μW power level. Consequently, the same superiority for energy consumption in WuR based solutions is observed in this study.

B. Performance of Scheme 2 as Device Population Varies

The performance of Scheme 2 is compared with a reference scheme, referred to as *Square*. According to this reference scheme, the region of interest is partitioned into multiple small square areas each covered by one WuC. To collect data, the UAV hovers only in the areas that have non-empty devices.

1) Access Success Probability: As illustrated in Fig. 8a), with a much larger population of 200 $\sim$ 700 devices, both schemes achieve very high probabilities when $L \geq 1.5N$. Moreover, *Square* achieves about 2 $\sim$ 5% higher success probabilities than Scheme 2. This is because *Square* divides the same region into more partitions leading to better coverage. However, the better coverage is obtained at a cost of longer access delay and higher energy consumption as presented below. In general, a joint performance assessment including all the three parameters is needed for proper parameter configuration.

2) Access Delay: For access delay in Scheme 2, it is worth mentioning that we consider only access delay as defined above, not the flying or/and hovering time for the UAV. It is evident from Fig. 8b) that our scheme achieves shorter delay for all three frame length configurations.

When a longer frame length is adopted, it will lead to longer delay. If we consider the delay performance only, it is preferable to configure a frame with the shortest length.

3) Energy Consumption: For energy consumption, it is evident from Fig. 8c) that lower energy consumption has been achieved when our scheme is employed regardless of the configured frame length.

For energy consumption with different frame lengths, we observe that a longer length leads to lower per-node energy consumption. By jointly considering the performance of all these three parameters, we would recommend configuring the frame length as $L = 1.5N$ for Scheme 2.

VI. Conclusions and Future Work

In this paper, we have proposed two receiver-initiated data collection schemes which target at providing collision-free data transmissions for WuR enabled small- and large-scale IoT networks respectively. Instead of waking up each device individually through a unicast WuC, we initiate a data collection procedure via a broadcast WuC to all devices in the region of interest and schedule their data transmissions via a hash function. Through analysis and simulations based on a static network topology, we demonstrate that the proposed schemes significantly improve the performance of traditional data collection schemes. As our future work, we will develop enhanced schemes dedicated for data collection in WuR enabled IoT networks with dynamic or ad hoc topologies.

References

[1] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled MAC protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 674-687, Apr. 2016.

[2] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, "Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 633-644, Apr. 2016.

[3] A. Frøytlog, T. Foss, O. Bakker, G. Jevne, M. A. Haglund, F. Y. Li, J. Oller, and G. Y. Li, "Ultra-low power wake-up radio for 5G IoT," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 111–117, Mar. 2019.

[4] F. Z. Djiroun and D. Djenouri, "MAC schemes with wake-up radio for wireless sensor networks: A review," *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 1, pp. 587-618, Q1, 2017.

[5] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 4, pp. 2117-2157, Q4, 2017.

[6] D. Ghose, F. Y. Li, and V. Pla, "MAC protocols for wake-up radio: Principles, modeling and performance analysis," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2294-2306, May 2018.

[7] L. Guntupalli, D. Ghose, F. Y. Li, and M. Gidlund, "Energy efficient consecutive packet transmissions in receiver-initiated wake-up radio enabled WSNs," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4733-4745, Jun. 2018.

[8] I. B. Damgård, "A design principle for hash functions," *Lecture Notes in Computer Science*, vol. 435, pp. 416-427, Jul. 2001.

[9] D. B. West, *Introduction to Graph Theory*, Pearson, 2015.

[10] J. Matoušek, *Lectures on Discrete Geometry*, Springer, 2002.