Geographical Correlation-based Data Collection for Sensor-augmented RFID Systems

Xin Xie, Xiulong Liu, Heng Qi, *Member, IEEE*, Bin Xiao, *Senior Member, IEEE*,
Keqiu Li, *Senior Member, IEEE* and Jie Wu, *Fellow, IEEE*

Abstract—This paper studies the practically important problem of data collection for sensor-augmented RFID systems. However, existing RFID data collection protocols suffer from two common limitations: execution time is naturally in proportion to the number of tags, thus they cannot satisfy time-stringent application scenarios; none of them is complaint with the C1G2 standard, thus they cannot be implemented using Commercial-Off-The-Shelf (COTS) RFID tags. To overcome these two limitations, this paper proposes the Geographical correlation-based RF-data Collection (GRC) protocol. GRC is fast because it is able to approximately capture the sensing data of all tags by only actually gathering data from a small set of sampled tags. This is based on the observation from the real-world data set that sensing data has a strong geographical correlation, *i.e.*, data gathered from nearby RFID tags has similar values. In GRC, we use a greedy approach to find the minimum sampling tag set to cover the whole monitoring region such that each un-sampled tag has at least one sampled tag nearby. Then, RFID reader runs the Framed Slotted Aloha (FSA) protocol specified in C1G2 standard to collect sensing data from the sampled tags. For each un-sampled tag, we approximate its sensing data by calculating weight-average of the data collected from its nearby sampled tags, where a faraway sampled tag should be given a small weight, and vice versa. Compared with existing RFID data collection schemes, the advantages of GRC are two-fold: (1) Extensive simulation results demonstrate that the time cost of our GRC scheme is only $1/28{\sim}1/3$ of the state-of-the-art data collection scheme; (2) GRC is totally complaint with C1G2 standard, thus it can be easily deployed on the COTS RFID tags.

Index Terms—Sensor-augmented RFID, Sensing Data Collection, Time-efficiency, Geographical Correlation.

1 Introduction

1.1 Background and Motivation

Radio Frequency Identification (RFID) has been widely used in various promising application scenarios such as supply chain management [1], [2], warehouse monitoring [3], [4], and inventory control [5], [6]. An RFID system typically consists of readers, tags, and a back-end server [7]. A tag is a microchip with an antenna in a compact package that has limited computing power and communication ranges. RFID tags can be classified into two types: *active tags*, which use the internal battery to power their circuits [8], and *passive tags*, which do not have their own power source and are powered up by harvesting the energy from the reader's electromagnetic fields [9]. The back-end server controls the RFID reader to send commands to query the tags, and the tags respond over a shared wireless medium [10]. Thus, information on tagged items can be automatically gathered.

With the development of chip manufacturing technology, RFID tags could be augmented with sensors [11], *e.g.*, WISP tags [12]. Thus, RFID tags can not only provide static ID information for inventory, but also real-time information about the state of the tagged object or the environment where these objects reside Compared with active tags and traditional sensors, WISP tags (a kind of sensor-augmented *passive tags*) have almost infinite lifetime due to battery-free manner, and thus more suitable for long-term monitoring purpose.

Information collection is a classical problem in sensor-augmented RFID systems. For example, in a large cold-chain storage facility, sensor-augmented RFID tags are attached to stacked food items and timely monitor their temperature. If abnormal temperatures of tagged items are discovered, we can take proper countermeasures to prevent from food spoilage. A great deal of efforts have been made to investigate how to gather exact information from a large number of sensor-augmented tags [13], [14]. However, time-efficiency of the existing solutions is still not satisfactory for tag-dense applications because they need to collect data of all tags. We observe from the real dataset [15] that, the sensing data measured by nearby tags has a strong correlation. For better time-efficiency, we propose to collect data from just a set of sampled tags (instead of all), and then use the collected data to approximate the data of un-sampled tags. Such a time-efficient solution with satisfactory data accuracy is preferred in many time-stringent applications.

1.2 Limitations of Prior Work

Due to its practical importance, a great deal of effort has been made by the academic community to address the problem of *RFID data collection* [14], [16], [17]. Chen *et al.* proposed a multi-hashing method called Multi-hash Information Collection (MIC) protocol [14], which makes use of the known tag IDs and the hash

- *Xin Xie and Heng Qi are with the School of Computer Science and Technology, Dalian University of Technology, China. E-mail: xiexindut@gmail.com, hengqi@dlut.edu.cn*
- *Keqiu Li is with the College of Intelligence and Computing, Tianjin University, Tianjin, China. E-mail: likeqiu@gmail.com.*
- *Bin Xiao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: csbxiao@.comp.polyu.edu.hk.*
- *Xiulong Liu is with the School of Computing Science, Simon Fraser University, Canada. E-mail: xiulong_liu@sfu.ca.*
- *Jie Wu is with the Department of Computer and Information Sciences, Temple University, USA. E-mail: jiewu@temple.edu.*

Corresponding authors: Xiulong Liu

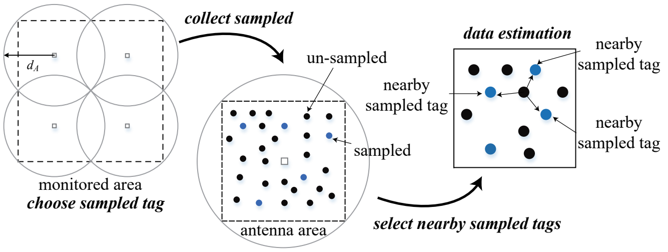


Fig. 1. The overview of geographical correlation-based data collection.

functions embedded in RFID tags to improve the frame utilization and avoid the transmission of tag IDs. Qiao *et al.* investigated how to efficiently collect information from a subset of given tags [17]. Yue *et al.* focused on the multi-reader RFID systems, and proposed to use the bloom filter to filter out the tag IDs that may reside in other reader's coverage [13]. A common objective of existing protocols is to alleviate the signal collisions among RFID tags to improve utilization of time frame, thereby achieving better time-efficiency. However, even if tag collisions are totally avoided, their ideal execution time is naturally in proportion to the number of tags, and thus cannot satisfy time-stringent application scenarios.

Although, statistical inferring-based data approximation problems have been extensively studied in the wireless sensor networks (WSN) [18], [19], the existing schemes cannot be directly used in sensor-argumented RFID system due to the following reasons. First, the RFID system is a single-hop network, and the sensor-augmented tags can only communicate with the reader but cannot communicate each other. On the contrary, the WSN is a multi-hop network consisting of multiple transmission links, and nearby sensor nodes can communicate each other in a distributed manner [20], [21]. Compared with WSN, the signal collision issue is much more serious in the RFID system because hundreds of tags simultaneously communicate with the single reader. Hence, the WSN protocols, which were mainly designed for distributed routing and networking, may not be able to handle such serious signal collisions in RFID systems. Moreover, sensor-augmented RFID tags can only harvest energy from the radio waves of reader, hence, their computation and communication capabilities are normally much weaker than battery-powered sensors. Hence, the complex WSN protocols [18], [22], which require heavy hardwares, usually cannot be applied on RFID tags. Due to the above two reasons, we need to propose designated protocols for RFID systems. We also proposed an error-bounded data collection protocol named SIC [23], which applies the sampled data to compute a fixed length interval that is expected to maximize the number of un-sampled tags. This straightforward method is of low accuracy because it uses a single value to approximate all the un-sampled tags.

1.3 Proposed Approach

We observe from a real-world dataset [15] that sensing data usually has strong geographical correlation, *i.e.*, data gathered from nearby sensing nodes within a certain distance usually has similar values. Making use of such geographical correlation, we propose the Geographical correlation-based RF-data Collection (GRC) scheme. At the beginning GRC, we carefully select a sampled tag set based on the pre-learned inherent data correlation, which ensures that each un-sampled tag can find at least one nearby sampled tag. A greedy algorithm is used to find a minimum sample set by giving preference to the tags with the maximum number of un-sampled nearby tags. Then, the reader issues Gen2 commands to gather sensing data from sampled tags as follows: The reader uses `Select` command to activate these sampled tags. Each `Select` command is embedded with a 96-bit ID, to activate the matched tags. Since tags communicate to the reader through a shared wireless channel, the reader issues `Inventory` command to resolve tag collisions. The `Inventory` command initializes a slotted time frame of f slots and each active tag randomly chooses a slot in the frame. The reader uses `QueryReap` commands to go through these slots one by one. Generally, there are three types of slots: *empty slot* in which no tag responds; *singleton slot* in which only one tag responds; *collision slot* in which two or more tags respond. RFID reader can only receive the sampled tags' data in singleton slots. The sampled tags in collision slots cannot successfully report their data, and will participate in subsequent inventory processes. Such a process will repeat until all sampled tags are successfully read. Next, the controller estimates the data on un-sampled tags using the weighted-average of multiple sampled data. The sensing data from nearer sampled tags are given higher weights because their data are more similar to that of the un-sampled tag. Moreover, a direction-based nearby tag selection algorithm and a compensation algorithm is used to preprocess these sampled data for improving the estimation accuracy. Our GRC scheme is much more time-efficient than the classical information collection scheme. The underlying reason is that, GRC only needs to gather information from a small set of sampled tags, and further makes use of the geographical correlation inherent in the sensing data distribution to approximate the data of un-sampled tags.

The time-efficiency of GRC is determined by the sample set, which may change round by round due to following three reasons. First, the distribution of sensory information across tags usually changes over time. For example, the distribution of temperature information in a room usually changes due to the varying sunlight. Since the optimal sample tag set is highly related to the information distribution, we need to dynamically adjust the sample tag set. Second, RFID tags may be associated with multiple kinds of sensors, e.g., temperature, humidity, and light. The results in Fig. 1 reveal that different kinds of sensory information have different geographical correlation. Hence, we usually need to use different sample tag sets for collecting different types of sensory information. Third, the tag population itself may also change over time, e.g., the tagged items are frequently moved out or in. In this case, we obviously need to adjust the sample tag set. When the sensing data share a very strong correlation, the sample set is usually small and GRC has a significant advantage over the information collection schemes, because the ratio of tags whose data can be approximated is large accordingly. However, this time-efficiency comes at the expense of losing some granularity when estimating the data of un-sampled tags. Fortunately, experimental results reveal that GRC normally involves a very small estimation error, which is acceptable for most RFID applications.

1.4 Technical Challenges and Solutions

We need to address the following three technical challenges when implementing the proposed GRC protocol. The first challenge is to design an effective algorithm to select a set of sampled tags with two objectives: First, it should minimize the number of sampled tags and guarantee that every tag can find at least one correlated nearby sampled tag within a certain distance. Second, it should return distinct sample sets at several consecutive GRC executions, such that all the tags have chance to report their sensory data. The proposed greedy sampling algorithm gives a higher sampling priority to the un-sampled tag that has more correlated neighbor tags.

The second challenge is to design an accurate estimator for un-sampled tags' data. In this paper, the un-sampled tags' data is estimated by the weighted-average of the nearby sampled tags' data. The closer sampled tag's data is assigned a larger weight because the sensing data from closer tags usually have a stronger correlation. Besides, we apply two techniques to improve the estimation accuracy. First, we only use the samples at different directions as the estimation input because these samples have more chance to compensate estimation error. Second, we propose a data-driven method to reduce the bias of the estimator by subtracting the expected deviation between the sampled and un-sampled data.

The third challenge is to ensure that our GRC approach is compliant with the EPC-Global C1G2 standard [24]. C1G2 compliant schemes are supported by various commercial RFID devices and can be easily deployed in current systems. Therefore, the reader communicates with the tag using C1G2 commands and avoids using any customized functions such as hash synchronization and vector comprehension.

1.5 Novelty and Advantage over Prior Art

This paper proposes a time-efficient RFID data collection scheme based on geographical correlation. Based on the real dataset [15], we investigate several factors that may affect the estimation accuracy, including distance, directions and original offset of the sensor nodes. By jointly considering all these factors, we design an efficient tag estimation scheme to use weighted-average estimator, direction-based filter and dynamic parameter updating techniques to guarantee the accuracy constrains. Compared with previous related works, the proposed GRC scheme has three major advantages. First, GRC leverages the geographical correlation and only needs to collect sending data from a set of sampled tags instead of all tags. Thus, the time-efficiency can be significantly improved. Second, in the classical statistical inferring approach [18], some sensing nodes may suffer from the "starving issue, *i.e.*, cannot be sampled all the time. In contrary, the proposed GRC protocol can ensure the fairness among tags, *i.e.*, each tag has a chance to be sampled and report its sensing data. Third, GRC is complaint with C1G2 standard, whereas previous RFID data collection protocols [14], [16] are not. Hence, GRC has better deployability. The simulation results reveal that our GRC scheme takes only $1/28 \sim 1/3$ of the execution time, compared with the state-of-the art information collection schemes. Table 1 summarizes the main notations used in this paper.

The rest of the paper is organized as follows. Section 2 introduces some preliminary knowledge about C1G2 standard. Section 3 presents the detailed design of our GRC approach. Section 4 discusses the communication cost, computation cost, and error control of GRC. Section 4 reviews the related work. Section 6 evaluates the performance of our GRC approach through extensive simulations. Finally, Section 7 concludes this paper.

2 Model & Assumptions

This section first presents the multiple access protocol used in GRC, and then presents the assumptions made in this paper.

TABLE 1
Key notations.

Notations	Descriptions
$\mathbb{I}/\mathbb{S}/\mathbb{U}$	Set of integrated tags; sampled tags; un-sampled tags
N_*	Size of set $*$, $*$ can be
Q	Parameter controls the length of time frame $f = 2^Q$
$X_{\mathbb{I}}$	Integrated data set $X_{\mathbb{I}} = \{i_1, \cdots i_{\lvert\mathbb{I}\rvert}\}$
$X_{\mathbb{S}}$	Sampled data set $X_{\mathbb{S}} = \{s_1, \cdots s_{\lvert\mathbb{S}\rvert}\}$
$X_{\mathbb{U}}$	Un-sampled data set $X_{\mathbb{U}} = \{u_1, \cdots u_{\lvert\mathbb{U}\rvert}\}$
$X_{\mathbb{E}}$	Evaluation data set $X_{\mathbb{E}} = \{e_1, \cdots e_{\lvert\mathbb{E}\rvert}\}$
$\mathbb{L}$	Locations of the tags
A_i	Covered area of antenna i
T_1/T_2	Waiting time on the reader/tag side
d	Distance threshold for geographical correlation
$d_{i,j}$	Distance between tag i and j
x_i, y_i	The x and y coordinates of tag i
k	Number of neighbor tags used for estimation

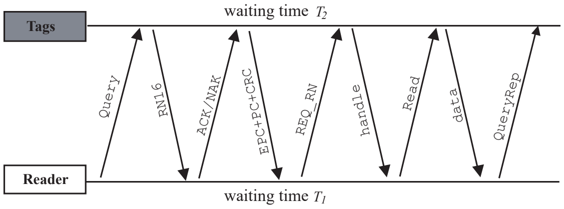


Fig. 2. Frame slotted aloha protocol specified in the C1G2 standard.

2.1 C1G2 Background

To be compliant with COTS RFID devices, we adopt the Q-protocol specified in C1G2 standard as the multiple access protocol. As illustrated in Fig. 2, the reader first issues `Query` command to start a frame of $f = 2^Q$ slots, where Q is an integer embedded in the `Query` command. Since $Q \in [0, 15]$, the maximum frame size is 32768. After receiving the `Query` command, each tag randomly chooses an integer ranging from 0 to f to initialize its slot counter. The tag whose slot counter equals to 0 needs to immediately respond to the reader by backscattering a 16-bit random number called `RN16`, which is used by the reader for tag verification in the subsequent process. Upon receiving `RN16`, the reader acknowledges the tag by sending an `ACK` command together with the received `RN16`. If multiple tags respond in the same slot, the reader cannot receive the correct `RN16` and send a `NACK` command to tags. If a tag receives an `ACK` command, it will respond with its 96-bit ID along with other information to the reader. Then, the reader issues a `Req_RN` command to gather information stored in the tag memory. The target tag responds with a 16-bit $handle$ for verification. Next, the reader issues the `Read` command embedded with the received $handler$ to read certain memory block stored on the tag. The tag with the verified $handler$ responds with the target data to the reader. At this point, the whole transaction cycle for information collection is done in this slot. The reader issues the `QueryRep` command to start the follow-up slot. After receiving `QueryRep` command, all tags decrease their slot counters by 1. Then, the tags whose slot counters become 0 will repeat the above transaction cycle. This process does not terminate until reader goes through all slots in the time frame.

2.2 Assumptions

RFID systems can be deployed in various ways to fulfill the needs of different applications. In this paper, we consider a scenario meets the following assumptions:

- *Controller:* We assume the central controller is connected with the RFID reader for conducting heavy computation tasks, generating reader commands on-line, and storing RFID data [14].
- *Antennas:* The RFID hardware part contains a reader that connects with multiple antennas [14], [25] to cover the monitoring region. We assume the antenna is idealized and all the antennas cover the same area of a circular region and are uniformly deployed at the region to cover the whole monitoring area. A batch of literatures have been proposed to study the reader antenna deployment problem [26], [27], *i.e.*, using the minimum number of antennas to seamlessly cover a specific monitoring region. Due to space limitation, we do not pay extra effort to study this problem any more.
- *Tags:* We assume the deployed RFID tags are sensor-augmented tags. Each tag is equipped with the same type of sensors to measure environmental information. The measured data is stored on the user block memory, which can be read by the reader through `Read` commands.
- *Locations:* The tags are attached to retail items for the tracking purpose. Since the objects can be placed anywhere when they are first moved into the system, their locations can be seen as random. Once these tagged items are deployed, we assume the tag locations are static and will not change during the monitoring time. The location of each tag can be obtained by executing the tag localization schemes [28], [29].
- *Data Correlation:* The geographical correlation naturally exists in a system where sensing nodes (*e.g.*, sensors or RFID tags) are densely deployed [18], [30]. A particular application has the specific geographical correlation. Before applying our proposed solution to an application scenario, we need to learn the inherent data correlation by jointly using the tag location information and the historical sensing data.
- *Time Stringent:* A long information collection process may disturb the execution of the other RFID application protocols, *e.g.*, tag localization [7], missing tag detection [31], and tag cardinality estimation [32]. Hence, we treat “time stringent” as a major concern in this paper, and desire a time-efficient information collection protocol.

3 Geographical Correlation-based Data Collection Protocol

In this section, we will present the proposed GRC approach in detail. We make extensive effort to optimize the parameters on both algorithm and command levels, thereby addressing the technical challenges specified in Section 1.4. The design goals of our GRC approach are three-fold: be complaint with C1G2 standard; satisfy the time-stringent application requirement; and guarantee the accuracy of un-sampled data. To achieve these goals, we propose a novel scheme that leverages the pre-learned inherent data correlation for estimating un-sampled data. Fig. 3 illustrates the GRC architecture. The three major stages including *sampling*, *estimating* and *updating* are specified as follows:

1) The controller applies a greedy algorithm to find a sampled tag set, which ensures that each un-sampled tag can find at least one nearby sampled tag and distinct tags should be sampled with the similar frequency.

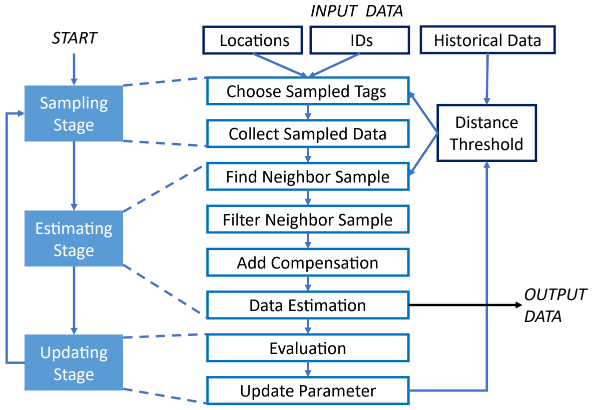
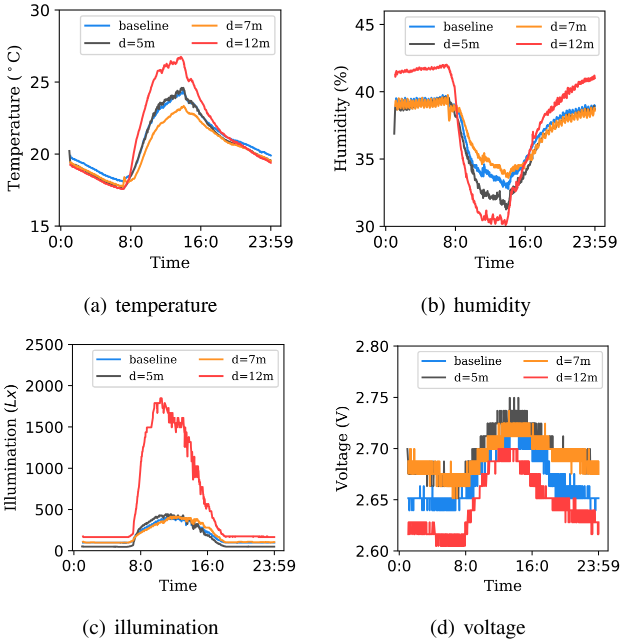


Fig. 3. Architecture of the proposed GRC scheme.



(a) temperature

(b) humidity

(c) illumination

(d) voltage

Fig. 4. Daily variations of four types of sensing data in one day

2) The RFID reader sequentially issues C1G2 commands to implement the selective reading for gathering sensing data from sampled tags.
3) The controller uses the weighted-average of multiple sampled data to estimate the un-sampled data. The data from nearer sampled tags are given larger weights. To improve the estimation accuracy, only the sampled tags in different directions of the un-sampled tag are chosen as the inputs of the weighted average estimator.
4) Finally, to evaluate the estimation accuracy, the controller collects the sensing data from some un-sampled tags and compares them with their estimation values. Based on the evaluation results, the controller will adjust the data correlation relationship to improve the time or accuracy efficiencies of GRC scheme.

3.1 Geographical Correlation between Sensing Data

GRC makes use of geographical correlation among sensing data to estimate the un-sampled data in sensor-augmented systems.

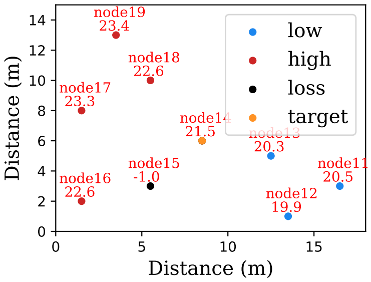


Fig. 5. Temperature data around node14

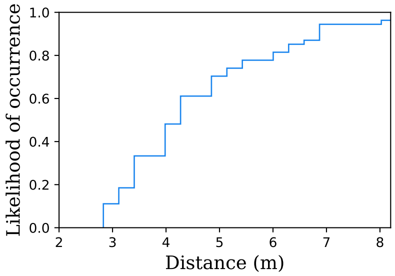


Fig. 6. Distribution of distance threshold

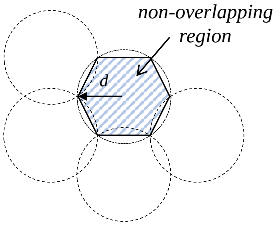


Fig. 7. Maximum non-overlapping

The geographical correlation naturally exists in a system where sensing nodes (*e.g.*, sensors or RFID tags) are densely deployed [18], [30]. A particular application has the specific geographical correlation related to surrounding environment, *e.g.*, the wall can break geographical correlation because two nearby tags separated by the wall may not have similar sensing data at all. Therefore, before applying our proposed scheme to an application scenario, we need to learn the inherent data correlation by jointly using the tag location information and the historical sensing data. The learned data correlation can be represented as an un-directed graph, in which each node is tag and a link means the two involved tags are close enough (their data are similar, and one data can be approximated by another). That is, we can dynamically learn the data correlation for a given RFID system.

In the following, we use an example dataset [15] to illustrate the geographical correlation between sensing nodes. The dataset contains four types of sensing data including temperature, humidity, illumination and voltage. We make three observations from Fig. 4. First, the distance is the major impactor factor on data correlation for all four types of sensor data. The numerical results in Fig. 4 coincide with this statement. According to Fig. 5, we can find that the direction is also a key factor that affects the geographical correlation of sensing data. Besides the factors of distance and direction, some external environment factors such as light and wind from air conditioner may also affect the geographical correlation. As exemplified in Fig. 4(c), the geographical correlation in illumination trace is strongly affected by the lighting condition. The illumination data of all nodes are of similar values because the dome light is off and all nodes have the same lighting conditions. These factors make sensor nodes usually follow the different distance-correlation relationship.

To compute the distance threshold for each individual tag, we evaluate the data from nearby tags based on user-defined accuracy metrics, such as the *maximum offset* or the *correlation coefficient*. No matter which metric is chosen, the methodologies for finding nearby tags are always same as shown in Algorithm 1. At the beginning, we compute the distance between arbitrary two tags. Then, we iterate each tag in set $\mathbb{I}$ to find its nearby tags, and use an undirected graph $\mathbb{V}$ to store the relationships of nearby tags. In $\mathbb{V}$, each node represents a tag in $\mathbb{I}$, and each link means two involved tags are nearby with each other. The distance threshold of the tag, is the distance to the farthest nearby tags. We use the dataset [15] as an example to investigate the distance threshold when determining the correlated nearby tags. Fig. 6 shows when the maximum temperature offset is set to 1 $^\circ C$, the distance thresholds are uniformly distributed between 3$\sim$8 meters at noon.

3.2 Sampling Stage

Given the distance threshold of each tag, we need a certain set of sampled tags $\mathbb{S}$ to make sure that each un-sampled tag can find at least one nearby sampled tag. The problem is to find a minimize size of sample set to cover all the un-sampled tags. This problem is similar to the *disk covering problem*, which have been studied extensively in wireless networks [33], [34]. However, RFID network has a different architecture. Thus, most of previous solutions cannot be cannot be applied in RFID networks. The sampling algorithm should return distinct sample sets at several consecutive GRC executions, such that all the tags can be sampled with similar frequencies. In the following, we will give two types of sampling algorithm meets the above requirements.

3.2.1 Random Sampling

Random sampling promises all the tag have the same chance for responding the reader. It is a naive scheme that brings little extra overhead. However, the performance of random algorithm is not satisfactory, because sampled tags chosen by random algorithm share many overlap regions. Besides, it can not ensure that the all the un-sampled tags are completely covered by the nearby sampled tags due to its probabilistic manner. For simplify, we can assume that each tag is randomly distributed within the region and each tag has the same covering region of $\pi d^2\ m^2$. The probability that an un-sampled tag is covered by at least one of the k sampled tags is presented as follows:

$$p \approx 1 - \left(1 - \frac{\pi d^2}{wh}\right)^k. \tag{1}$$

The required sample size to cover p of the tags is:

$$k_r = \frac{\log(1-p)}{\log(wh - \pi d^2) - \log(wh)} \tag{2}$$

To implement random sampling scheme on Gen2 devices. The reader only needs to issue `Inventory` commands to start an identification cycle using the frame slotted Aloha protocol. The tags in singleton slots can be successfully identified by the reader. Fowling each successful identification, the reader will issue `Read` command to collect information stored on the tags. The reader stops the above process until the required number of sampled tags is identified. Let t_s, t_e, t_c and t_{data} denote the time length of singleton slots, empty slots, collisions slots and data collection slots, respectively, the total communication overhead of random sampling scheme can be approximated to $T_R = |\mathbb{S}_R| \times (t_e + t_s + (e-2)t_c + t_{data})$, where $|\mathbb{S}_R|$ represents the number of sampled tags selected by the random algorithm.

Algorithm 1: Nearby tag detection algorithm

```
Data: tag locations $\mathbb{L}$ and historical data $X_{\mathbb{I}}$
Result: data correlation graph $\mathbb{V}$
initialize an undirected graph $\mathbb{V}$ of $|\mathbb{I}|$ nodes;
for each tag $x$ in $\mathbb{I}$ do
    sort tags in $\mathbb{I}$ based on their distance to $x$;
    for each tag $y$ in sorted $\mathbb{I}$ do
        if $x$ and $y$ has geographical correlation then
            add a link between $x$ and $y$ in $\mathbb{V}$.
        else
            break;
        end
    end
end
```

Algorithm 2: Greedy sampling algorithm

```
Data: undirect graph $\mathbb{V}$ storing the nearby tags;
      un-sampled tag set $\mathbb{U}$
Result: sampled tag list $\mathbb{S}$
while $\mathbb{V} \neq$ empty do
    iterate nodes in $\mathbb{U}$;
    tag $x$ = node with the maximum number of links;
    add $x$ to sample list $\mathbb{S}$;
    remove tag $x$ from $\mathbb{U}$ remove nodes that have a link to
      $s$ from graph $\mathbb{V}$;
    remove node $x$ from $\mathbb{V}$.
end
```

3.2.2 Greedy Sampling

Alternatively, we propose a greedy algorithm to select a set of sampled tags with two objectives. First, it should minimize the number of sampled tags and guarantee that every tag can find at least one correlated nearby sampled tags within a certain distance. Second, it should return distinct sample sets at several consecutive GRC executions, such that all the tags can be sampled with similar frequencies. The detailed algorithm is presented in Algorithm 2. We use $\mathbb{U}$ to denote the set of un-sampled tags, which is initialized to $\mathbb{I}$ at the very beginning of GRC. $\mathbb{V}$ is an un-directional graph used to represent the data correlation of the system, and its initialization value can be obtained from the methods in Section 3.1. Then, we will iterate each node in $\mathbb{U}$ to find out a node x with the maximum number of links. Such a node is equivalent to the tag that has the most nearby tags, which should be added to the sample tag set $\mathbb{S}$ and removed from the un-sampled tag set $\mathbb{U}$. The tags that have links to tag x are the nearby tags, whose data are correlated to that of tag x. Thus tag x, its nearby tags, and their links should be removed from $\mathbb{V}$. After that, the controller again iterates each node in the updated $\mathbb{U}$ to find the new node x' with the maximum number of links, and repeats the above processes to updating $\mathbb{U}$ and $\mathbb{V}$. Such a process is repeated until $\mathbb{V}$ becomes empty. To start the next round of information collection, $\mathbb{V}$ is reset to its initialization value again. For fairness concern (*i.e.*, letting each tag has a chance to be sampled and report its sensing data), we need to reset $U = I$ when U becomes empty. Fig. 9 uses the example dataset [15] to show a set of sampled tags selected by the greedy algorithm.

To collect information from a certain set of sampled tags with Gen2 devices, we need to implement a selective reading with the Gen2 commands. First of all, the reader issues `Select` commands embedded with tag IDs to active all the sampled tags in $\mathbb{S}$. Then, the issues `inventory` command to starts a time frame to identify the active tags using frame slotted Aloha. The tags in singleton slots can be successfully identified by the reader. Following each successful identification, the reader will further issue `read` command to collect information stored on the identified tag. Let t_p denote the overhead of polling tag with the ID, the total overhead of selective reading is $T_G = |\mathbb{S}_G| \times (t_p + t_s + t_e + (e-2)t_c + t_{data})$, where $|\mathbb{S}_G|$ represent the number of sampled tags selected by the greedy algorithm.

3.2.3 Performance Comparison

Obviously, the greedy algorithm returns a smaller sample set compared to the random algorithm because we always select the sampled tag that has the maximum number of neighbor tags. What's more, greedy algorithm is a deterministic method that ensures all the un-sampled tags can be covered by at least one nearby sampled tag while uniform random algorithm is a probabilistic method that never provides guarantee on completely coverage. We conduct a set of simulations to compare the performance of the proposed greedy algorithm, the random algorithm and the optimal lower bound. In the simulations, we assume that the monitoring region is a $17 \times 17m^2$ square area, where 1000 tags are uniform-randomly deployed. As shown in Fig. 7, suppose each tag can cover a πd^2 circle region, the non-overlapping area covered by each tag is a hexagon with the side of d. Therefore, given a $w \times h$ rectangle, the optimal lower bound on sampled tags should be larger than $k_o = \frac{2\sqrt{3}wh}{9d^2}$. As shown in Fig. 8, we can find that the greedy algorithm always returns a smaller sample tag set compared with the random algorithm. For example, when $d = 3m$, the size of sample tag set returned by the greedy algorithm is only one third of the random sampling scheme (99.9%), and is twice the optimal lower band. Since the communication cost of both random algorithm and greedy algorithm is a function of $|\mathbb{S}|$, where $|\mathbb{S}|$ is the size of sample tag set. We can assert that the proposed greedy algorithm is much more time-efficient than the straightforward random algorithm. Besides, the gap between the random algorithm and our greedy algorithm becomes larger as the increase of coverage rate. For example, when $d = 4m$, the size of sample tag set for covering 99.9% of the tags is 3 times that for covering 95% of tags.

3.3 Estimating Stage

Given the sampled data $X_{\mathbb{S}}$, the next issue is how to apply them to estimate the data of un-sampled tags. We have made the following two observations based on the practical dataset [15]. First, as shown by the orange line in Fig. 10, the tags closer to sampled tags are expected to have smaller absolute errors. Thus, when estimating the sensing data of an un-sampled tag, we need to give larger weights to the sampled tags that are closer to this un-sampled tag and use the weighted-average value as its estimation data. Second, as shown by the blue line in Fig. 10, the errors can be either negative and positive over time. Therefore, the tag tags far from the sampled tags may have small average errors because the above two types of errors can cancel with each other. The underlying reason is that geographical distribution of many kinds of environmental data can be approximated by the *multivariate normal distribution* [35], [36], which generally has opposite gradient values at opposite directions. Therefore, we

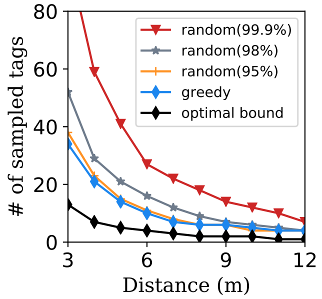


Fig. 8. Samples size comparison

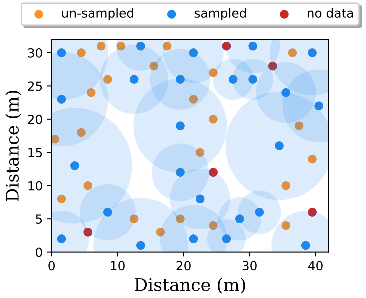


Fig. 9. Greedy sampling result

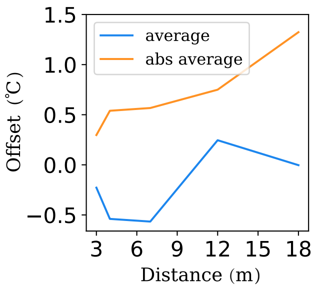


Fig. 10. Avg. offset *vs.* distance

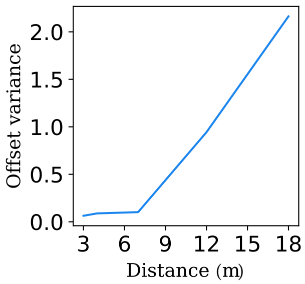


Fig. 11. Variance *vs.* distance

prefer to choose samples along different directions when there are multiple sample tags. Therefore, when estimating the sensing data of an un-sampled tag, we prefer to choose the nearby sampled tags at different directions. For example, in the dataset corresponding to Fig. 4(a), the sampled tags with $d = 7m$ and $d = 12m$ are at different directions to the un-sampled tag. As shown in Fig. 4(a), we can achieve a better estimation result for this un-sampled tag if averaging the sensing data of these two sampled tags than separately using either of them. Based on the above observations, we estimate the un-sampled data by weighted-averaging the data from multiple sampled data in different directions.

In the following, we present an algorithm to select sampled tags whose data are used as the input of weighted-average estimator. First, the sampled tag nearer to the un-sampled tag are sampling with high priority. Second, when selecting a new tag, we must ensure that the degree between the new tag and any selected tags exceeds or equals to 90 degrees. Let (x_i, y_i), (x_j, y_j) and (x_t, y_t) denote the location of sampled tag i, sampled tag j and un-sampled tags t, respectively. The cosine for the angle between tag i and tag i is:

$$\cos\alpha_{ij} = \frac{(x_i - x_t)\cdot(x_j - x_t) + (y_i - y_t)\cdot(y_j - y_t)}{\sqrt{(x_i - x_t)^2 + (y_i - y_t)^2} + \sqrt{(x_j - x_t)^2 + (y_j - y_t)^2}}.$$

$\alpha_{ij} > 90^o$ if $(x_i - x_t)\cdot(x_j - x_t) + (y_i - y_t)\cdot(y_j - y_t) < 0$. Fig. 12 shows an example of the filtering process. Tag i is selected because it is the nearest tag to the un-sampled tag; tag j is selected because $\alpha_{ij} > 90^o$; but tag w is not selected because $\alpha_{iw} < 90^o$.

After removing sampled data at similar directions, we apply a data driven approach to reduce the initial offset between the un-sampled tag's data and selected sample tags' data. Let $\hat{u}_{ij}$ denote the estimator of un-sampled tag i based on sampled tag i, we have:

$$\hat{u}_{ij} = u_j + E(u_i - u_j), \tag{3}$$

where u_* represents the data value of tag $*$ and $E(u_i - u_j)$ denotes the expected difference between the un-sampled data u_i and the sampled data u_j. $E(u_i - u_j)$ can be computed by averaging the historical value difference between u_i and u_j during a certain time like one day as follow:

$$E(u_j - u_i) \approx \sum_{t=1}^{T}\left(u_j^t - u_i^t\right)/T. \tag{4}$$

where T denotes the length of the historical data sequence and u_*^t denotes the sensing data of tag $*$ at time t.

Finally, we use the weighted average of the compensated data to estimate the un-sampled data. Since the estimation variance of closer tags are smaller as shown in Fig. 11, we assign a larger weight coefficient to the data of sampled tag which is closer to the un-sampled target. Specifically, the weight coefficient w_{ij} is:

$$w_{ij} = \frac{1/d_{ij}^2}{\sum_{j=1}^{k} 1/d_{ij}^2}, \tag{5}$$

where d_{ij} denotes the distance between the target tag i and its nearby sampled tag j. The final estimation equation for estimating the data of un-sampled tag i can be expressed as follow:

$$\hat{u}_i = \sum_{j=1}^{k} w_{ij}\cdot\hat{u}_{ij}, \tag{6}$$

3.4 Updating Stage

In this section, we present how to update the distance thresholds with the change of time. Since the geographical distribution of the measured parameters is dynamic and changes during the GRC executions, we need to update distance thresholds to catch up the up to date correlation relationship between tags. Fig. 4(a) shows that the offset between sensors is relatively large in the middle of the day but turn to small at night. Updating the threshold d in time is a key factor to improve both time and accuracy efficiencies of GRC scheme. The updating strategy is chosen based on the evaluation results on estimation accuracy. To evaluate the estimation accuracy for un-sampled tags, besides collecting sensing data from the sampled tags, the reader needs to collect the sensing data from some un-sampled tags. Let $\mathbb{E}$ denote the set of selected un-sampled tags used for evaluation of estimation accuracy. For each un-sampled tag in $\mathbb{E}$, we will compare the offset between its exact data e_i and the estimated value $\hat{e}_i$. If the offset exceeds a threshold specified by the user, the correlation link between x and its nearby sampled tag y should be removed. Moreover, the correlation links to the sampled tags, which are farther than the removed sampled tag y, will be also removed from the nearby tag set of tag x. Otherwise, we will add a correlation link between x and its nearest un-linked tag to reduce the size of sampled tag set for better time-efficiency. If there are too many estimation errors, we need to remove all correlation links and re-compute all the distance threshold for each tag.

4 Discussion on Practical Issues

This section first analyzes the communication and computation overhead of GRC scheme, and then present why we choose Frame Slotted Aloha protocol specified in C1G2 [24] as the MAC layer communication protocol.

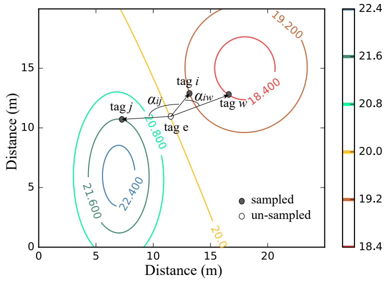


Fig. 12. An example of temperature distribution.

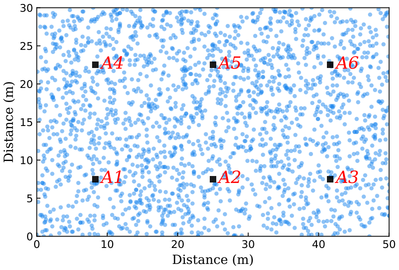


Fig. 13. The geographical tag distribution.

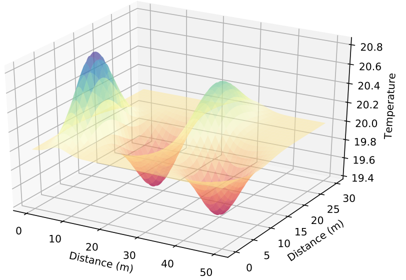


Fig. 14. The geographical data distribution.

4.1 Communication Cost

Reading data from sampled tags is the major communication cost of GRC. Let N_A be the number of RFID antennas deployed in the monitoring region, and m be the number of sampled tags within the communication range of each antenna, the total number of sampled tags can be denoted as $N_{\mathbb{M}} = m \cdot N_A$. By contrast, the standard information collection protocol needs to collect data from $N_{\mathbb{I}}$ integrated tags, which incurs much more communication and time cost. Since the communication cost is in proportion to the number of collected tags, the communication costs reduced by GRC is $(N_{\mathbb{I}} - N_{\mathbb{M}})/N_{\mathbb{I}}$. The improvement increases as $N_{\mathbb{M}}/N_{\mathbb{I}}$ becomes smaller. While $N_{\mathbb{M}}$ is in proportion to the area of the monitoring region, we can conclude that GRC has a more significant improvement when the tags are deployed with a higher density. This is reasonable because each sample can cover so many un-sampled tags in high density deployment. According to our observation on temperature sensing dataset, 20 samples is enough for the monitored area of $200m^2$, GRC can reduce over 90% cost when the number of tags deployed within this region is larger than 200, which is very common in inventory and warehouse applications. Therefore, we believe that the proposed GRC can significantly improve the time-efficiency than existing exact data collection approaches.

4.2 Computation Cost

GRC significantly reduces the communication cost of the information collection operation at the expense of involving some extra computational cost. The computational cost of GRC mainly contains three parts: tag sampling, un-sampled tag estimation and distance threshold updating. The overhead of greedy sampling algorithm is $O(|\mathbb{I}|^2 \log |\mathbb{I}|)$. The overhead of un-sampled tag estimation can be represented as $O(T|\mathbb{U}|)$, where T denotes the length of historical data. The overhead of distance threshold updating can be represented as $O(|\mathbb{E}|)$. Since $|\mathbb{E}| \ll |\mathbb{U}| < |\mathbb{I}|$, the greedy sampling algorithm accounts for the majority of computational costs of GRC and the total complexity of GRC can be represented as $O(|\mathbb{I}|^2 \log |\mathbb{I}|)$, which can be quickly accomplished in a normal computer. Thus, similar with most RFID literature [37], the computation time involved in GRC is ignored because it is negligible compared to the communication time. For fair comparison, we also do not consider the computation time of the benchmark approaches when comparing their performance.

4.3 Impact of Channel Errors

In real-world environment, the communication channel is usually error-prone, because white noise may corrupt the message exchanged between the reader and tags, *e.g.*, 0 becomes 1 or 1 becomes 0, causing bit error. More seriously, some messages are even not detected at all due to the path loss. Most of the literature focuses on minimizing the transmission bits or execution time of the protocol. They usually adopt a time-efficient data structure, *e.g.*, Bloom filter, and assume the data structure can be correctly received by all tags in the system. However, it is a stringent requirement due to the unavoidable channel errors. Therefore, GRC uses Frame Slotted Aloha protocol specified in C1G2 [24] as the MAC protocol, which designs a sequence of mechanisms for handing transmission errors. Compared with existing RFID data collection protocols, our GRC protocol provides a more reliable transmission by exploiting various fault-tolerant mechanisms such as well-designed state transform model and Cyclic Redundancy Code (CRC). Since each data is packaged with CRC code, the receiver (a tag or a reader) can detect the bit error of the message by CRC verification. If it fails to pass the CRC verification, the whole message is dropped. The receiver considers it receives an invalid command and follow the state transformation defined in [24]. If the receiver is a tag, it reselects a slot and waits for the subsequent commands; if the receiver is a reader, it terminates the current slot and starts a new slot. On the other hand, if the receiver does not receive the message after a period, it resets its state which is similar to the actions after receiving an invalid command.

5 Related Work

RFID tags contain various types of concerning data, including ID, status code, information in tag memory, and even the data sensed by the embedded sensor. One of the most fundamental tasks in RFID research is to design a scheme for efficiently reading these data from a large batch of tags. ID reading is the hottest topics in the early stage of RFID research, the major challenge is how to resolve signal collisions among tags when the reader interrogates these tags. The prior works on anti-collisions problem can be classified into two categories: Aloha-based [38]–[40] and Tree-based [41], [42]. The Aloha-based protocols can be interpreted as a kind of Time Division Multiple Access (TDMA) mechanisms. The reader sends a value f to the tags in its interrogation range where f indicates the number of slots in the forthcoming time frame. Then, each tag randomly picks a time slot in the frame to respond to the reader. If one and only one tag responds in a time slot (this slot is called singleton slot), the reader can successfully identify this tag. If two tags simultaneously respond in a slot (this slot is called collision slot), the reader cannot derive any tag IDs due to signal corruption. The unidentified tags will participate in the next frame. Such an iterative identification process will not terminate until all the tags are identified.

The Tree-based protocols are also a kind of fundamental multiple access protocols, which are first invented by U.S. Army for testing soldiers for syphilis during World War II. The basic idea is that the reader first queries 0 and all the tags whose IDs start with 0 respond with their IDs. If the reader successfully identifies a tag (*i.e.*, only one tag responds) or just reads an empty slot (*i.e.*, no tag responds), it turns to query 1 and all the tags whose IDs start with 1 respond. In contrary, if the reader receives a collision, which means that there are two or more tags whose IDs start with 0, it generates two new query strings ***0 and ***1 by appending a 0 and a 1 to the previous query string ***. Then, the reader sequentially uses these two strings to query the tags. This process continues until all the tags have been identified. Fundamentally, the Tree-based protocols can be interpreted as depth-first-search query mechanisms. Due to its simplicity and practicality, Q protocol, a Aloha-based protocol, is specified as the MAC protocol of C1G2 standard [24].

As sensor-augmented RFID tags have been increasingly adopted in various application scenarios, it is practically important to design a time-efficient approach to gather the sensing data. The data approximation problem has been extensively studied in wireless sensor network literature, which usually leverage the spatial and temporal correlation between sensor nodes to improve the time and energy efficiencies of information collection [18], [19]. For example, EEDC [43] partitions sensor nodes into clusters based on their data similarity. Hence, the observation at any point of the cluster can be approximated by the observation of any nodes in the cluster, which significantly reduce the overhead to report sensing data. On the other hand, Ken [18] uses replicated dynamic probabilistic models to minimize the data collection overhead from sensor nodes to PC. All the sensor nodes share a data prediction model with the PC and knows whether the prediction result is right or not. Hence, the sensor node only reports its data to the controller if the value predicted by the probabilistic model is wrong. Moreover, some most recent scheme also divides into the physical layer and investigate how to leverage the low-level link correlation to reduce the communication overhead [44]. They cannot be directly applied in RFID systems because RFID network has a different architecture and the RFID applications always have different requirements. For example, the RFID tags cannot communicate with each other and every sensor-augmented tag should have the same chance to report their data. This motivates us to design a data estimation scheme GRC for the RFID system.

In recent years, many researchers focused on collecting sensing information rather than just identifying IDs. Collecting information with frame-slotted Aloha is of low time-efficiency, because the utilization ration always below 37%. Therefore, most of prior work on information collection focus on resolving tag collisions for improving the utilization ration of time frame. Chen *et al.* proposed a Multiple Hash information Collection protocol (MIC) in [14]. MIC improves the utilization ration to 80% by combining multiple hash functions to select the slot. With multiple hashing functions, the tags map to collision slots with the first hashing, have a chance to be mapped to singleton slots with the other hashing. The reader needs to send a message to inform tags of the hash functions they adopted. To further accelerate the information collection in multi-reader RFID system, Zhang *et al.* [16] proposed a Bloom filter-based protocol (BIC) to identify tags in the region of each reader, then the reader can work in parallel for gathering sensing data from tags within its own region. Although the above solutions improve the time-efficiency to some extent. The time cost is still in proportion to the number of monitored tags, which cannot satisfy time-stringent application scenarios. Besides, these solutions are not complaint with C1G2 standard because customized functions and additional communication stages are involved. Thus, none of previous RFID data collection protocols can be used in practice.

6 Performance Evaluation

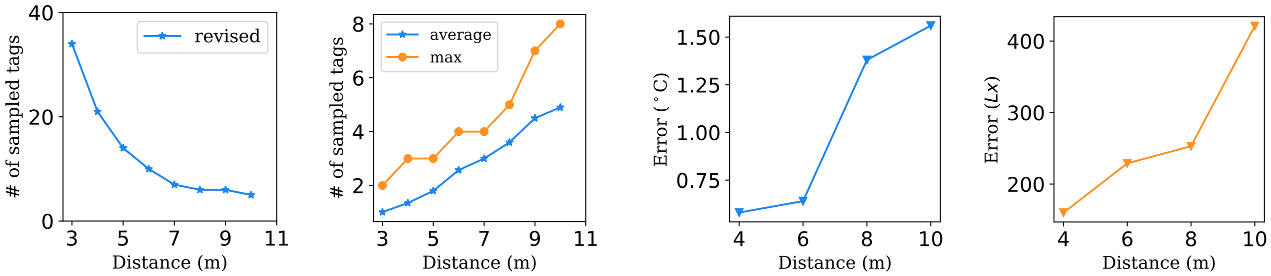
In this section, we simulate GRC in python and evaluate its performance through both on a practical dataset [15] and simulated data. We investigate both the data collection accuracy and time-efficiency of GRC under various parameters settings. We also implement three representative RFID data collection algorithms, *i.e.*, Gen2 [24], MIC [14] and BIC [16], to compare their performance with GRC side by side.

6.1 Experiments Study

First of all, we evaluate the accuracy of the proposed scheme on a practical dataset which is collected from 54 sensing nodes deployed in the Intel Berkeley Research lab [15]. We investigate the number of sampled sensing nodes, cluster size and two types of estimation error with varying unified threshold distance d. As shown in Fig. 15(a), the number of sampled nodes selected by the greed algorithm decreases with the increase of distance. Since GRC only gathers sensing data from the sampled nodes, the communication cost can be significantly reduced. Each sampled node and the nodes whose data approximated by this node can be seen as a cluster. The average cluster size can be regarded as the speedup rate of GRC, which significantly increases with the increase of distance threshold. When evaluating the estimation error of a tag, we compare the average offset between the exact value and its estimation value within a day. We use light dataset and temperature dataset to conduct simulations to evaluate the performance of the proposed GRC scheme. The simulation results are shown in Figs. 15(c) and 15(d), respectively. It is easy to observe that, estimation errors of both temperature and illumination increase with respect to d. We also observe an interesting phenomenon: there are sharply increased errors at some locations. This is because the similarity between two sensing nodes can be strongly affected by the structure of the indoor environment. For example, two nearby nodes separated by the wall may not have strong correlation. With the increase of distance thresholds, sensing nodes at different rooms may be assigned into the same cluster, resulting in sharply increased estimation errors. In conclusion, the experiment study shows that GRC can achieve approximate data estimation with a suitable distance threshold d. To further improve the estimation accuracy, we need to take many other environmental factors into consideration, *e.g.*, room layout, time and the facility that may affect the measured parameters.

6.2 Simulation Settings

In our simulations, the monitoring area is a $50m \times 30m$ rectangle as illustrated in Fig. 13, where 2000 sensor-augmented UHF tags are randomly deployed. To cover such a region, 6 RFID reader antennas $[A1, A2, A3, A4, A5, A6]$ with the communication range of $12m$ are uniformly deployed on the monitored area, and the distance between any two reader antennas is $12\sqrt{2}m$. Let the left-bottom point of the monitored area be the origin point (0,0), the position of each tag is randomly generated with a granularity of $0.01m$. In Fig. 13, each colored dot denotes a RFID tag and the black squares denote the locations of reader



(a) # of sampled tags (b) cluster size *vs.* distance (c) temperature *vs.* distance (d) illumination *vs.* distance

Fig. 15. Performance evaluation based on Intel Lab dataset [15]

TABLE 2
Communication setting

Collisions	Bits	Time
Query	22 bits	$0.55ms$
RN16	16 bits	$0.4ms$
QueryRep	4 bits	$0.1ms$
QueryAdjust	9 bits	$0.225ms$
ACK	18 bits	$0.45ms$
NAK	8 bits	$0.2ms$
REQ_RN	40 bits	$1ms$
GRC	16 bits	$0.4ms$
Read	58 bits	$1.45ms$
data	49 bits	$1.225ms$
EPC+GRC+PC	128 bits	$3.2ms$

TABLE 3
Time cost for slots

Type	Bits	Time
Collision	50 bits	$1.825ms$
Empty	50 bits	$1.825ms$
Successful	345 bits	$9.925ms$

antennas. Besides, we use a mixture bivariate normal model to simulate the measured temperature data in the monitoring area. The detailed temperature at each point are shown in Fig. 14.

The communication parameter settings follow the specification of the Gen2 standard [24]. We assume the length of sensing data is 16-bit long. Data transmission rate between reader and tag is equivalent, both $40kb/s$, *i.e.*, it takes $25us$ to transmit one bit. All the related commands and its transmission time are shown in Table 2. Besides, let T_{pari} be the backscatter-link pulse-repetition interval, the waiting time between reader transmission and tag response, and the waiting time between tag transmission and reader response are $T_1 = 10T_{pari}$ and $T_2 = 3T_{pari}$, respectively. Because $T_{pari} \approx 25us$, we have $T_1 = 250us$, and $T_2 = 75us$. Based on Table 2, we can obtain the time cost of each type of slots as shown in Table 3.

6.3 Evaluating the Estimation Accuracy

In the simulations, we apply GRC to collect equality number of sampled tags from each antenna region simultaneously and independently and combine them as the input of the proposed data estimation algorithm. We assume there is no historical data, thus the expected offset between any un-sampled tag and sampled tags is set to zero.

Figs. 16, 17 and 18 show the estimation accuracy with varying number of sampled tags. We observe that the mean estimation offset becomes smaller and smaller when there are more sampled tags. This is easily interpreted because with more sampled tags, an un-sampled tag is expected to find closer sampled tag, whose sensing data is highly correlated to the sensing data of this un-sampled tag. On the contrary, the maximum estimation offset is not sensitive to the number of sampled tags, and almost keeps unchanged. This is because GRC is hard to estimate the un-sampled tags located at the peak points of data distribution map shown in Fig. 14. The underlying reason is the gradient around the peak point is extremely large, resulting a large estimation error. To accurately estimate the peak point value, the sampled tag should be very close to the peak point, which requires an extremely large number of sampled tags and lower the speedup rate of GRC.

Another interesting observation from Figs. 16, 17 and 18 is that different reader antenna regions have different estimation accuracy. For example, the estimation accuracy in $A1$ is worse than other regions. This is because the measured environmental data have distinct geographical distribution in different A_i. By jointly considering Fig. 14 and Fig. 13, we can find that the $A1$ region has a significant temperature change, thus resulting in larger estimation offsets. In conclusion, the estimation offset is affected by distribution of environmental data (*e.g.*, temperature distribution). In the region with drastic data changes, we need to set a smaller threshold distance and involve more sampled tags for ensuring estimation accuracy.

Fig. 19 presents the cumulative histogram of estimation offset. When there are 20 sampled tags, more than 70% tags have an estimation offset smaller than 0.1. By contrast, when there are 5 sampled tags, only 50% tags have an estimation offset smaller than 0.1. When using GRC, we need to trade-off between the time-efficiency and estimation accuracy. If we prefer a high estimation accuracy (*e.g.*, offset < 0.1) with a high reliability, we need to select more sampled tags; otherwise if certain error is acceptable, we can accelerate GRC by collecting data from just a small set of sampled tags. Finally, we investigate the relationship between estimation offset and distance. From Fig. 20, we can see that the estimation offset increases as the distance from the nearest neighbor increases. We can use a linear function $o = 0.02 \times d + 0.02$ to fit the relationship between distance d and the offset o, which shows the estimation offset is in proportion to the distance between an un-sampled tag and sampled tags.

6.4 Evaluating the Estimation Algorithm

In this section, we conduct simulations to compare two kinds of estimation algorithms: nearest neighbor algorithm and

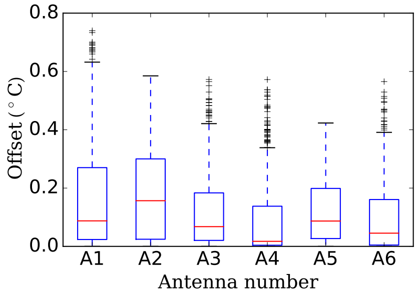


Fig. 16. Estimation offset: 5 sampled tags.

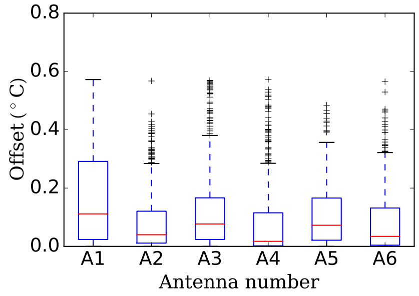


Fig. 17. Estimation offset: 10 sampled tags.

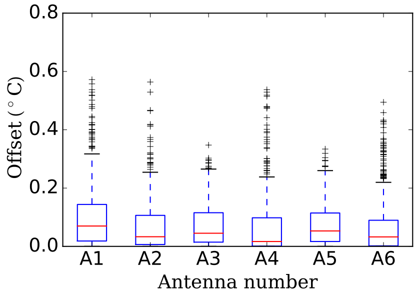


Fig. 18. Estimation offset: 20 sampled tags.

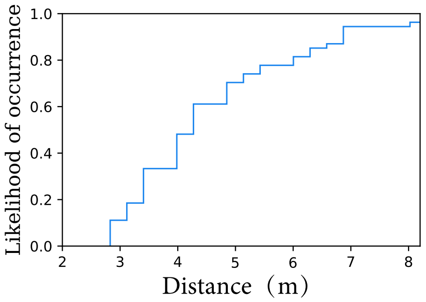


Fig. 19. Cumulative estimation offset.

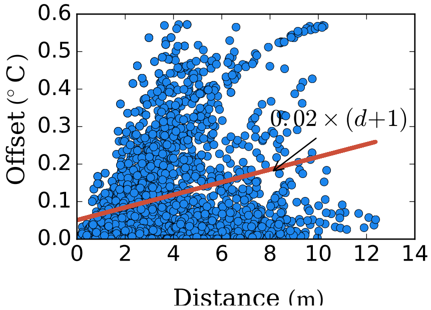


Fig. 20. Correlation between distance&offset.

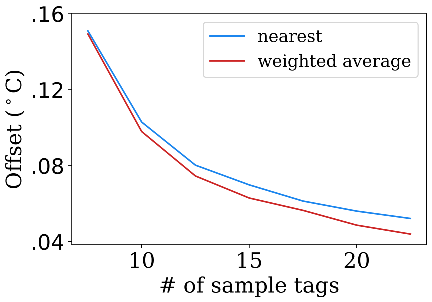


Fig. 21. Comparison of two average methods.

TABLE 4
Comparison of related protocols in execution time (seconds)

$N_{\mathbb{I}}$	C1G2	MIC	BIC	GRC ($s = 10$)	GRC ($s = 20$)
1K	14.67	9.60	8.61	1.32	2.66
3K	44.57	28.79	25.82	1.33	2.71
5K	74.59	47.98	43.04	1.31	2.68
7K	103.58	67.18	60.26	1.34	2.70
9K	134.94	86.37	77.47	1.30	2.68

weighted-average algorithm. The nearest sample estimator approximates the data of un-sampled tags with the data from its nearest sampled tag; whereas the neighbor weighted average algorithm presented uses weighted average of several sampled tags to estimate the data of un-sampled tag. Since our simulations are conducted in a static case without time variant features, we assume there is no historical data and the expected compensation $E(u_j - u_i)$ in Eq. 3 is set to 0. As shown in Fig. 21, even without adding compensation, the neighbor weighted-average algorithm always provides a more accurate estimation compared to the nearest sample estimator. The gaps are getting larger as the number of sampled tags increases. The underlying reason is that the neighbor weighted-average algorithm has more chance to eliminate the estimation offset by balancing the positive and negative sample estimators.

6.5 Evaluating the Time Efficiency

In this section, we compare the execution time of GRC with recent whole-set information collection protocols including C1G2 [24], MIC [14] and BIC [16]. First, we assume the monitoring area is a triangle area of $50 \times 30/m^2$ covered by 6 reader antennas, and vary the number of tags from $1K$ to $9K$. We run GRC at two different settings: GRC ($s = 10$) collects 10 sampled tags from each antenna region; GRC ($s = 20$) collects 20 sampled tags from each antenna region. We run each protocol 300 times to get the average execution time. The results in Table 4 show that GRC significantly outperforms the state-of-the-art BIC in terms of time-efficiency. For example, the execution time of GRC ($s = 10$) is only 15.3% of the time cost of BIC. As the number of tags increases to $5K$, the gap becomes larger, where GRC only costs 0.3% of the time cost of BIC. Such huge improvement lies in the following reasons. GRC only needs to collect data from a small set of sampled tags, which accelerates the information collection by significantly reducing the execution time. Although it also brings additional computational overhead, the computation time is negligible as we have mentioned in Section 4.2.

Another observation is the execution time of all prior works are in proportion to the number of tags. It is because prior methods need to collect data from all tags. By contrast, the execution time of GRC keeps stable because its execution time is related to the geographical distribution of the measured physical parameters as well as the required accuracy. Due to this property, we can conclude that GRC has a better performance than previous protocols in the applications where tags are densely deployed. Because the execution time and estimation accuracy of GRC is not related to the density of tags. In an application with high tag density, prior protocols take too much time to collect data from so many tags, which is of low time-efficiency and may disturb other RFID operations. In the contrary, GRC can significantly accelerate the information collection at the expense of introducing slight estimation offset as shown in Figs. 16, 17, 18.

7 Conclusion

This paper makes the following contributions. First, we propose a Geographical correlation-based RF-data Collection (GRC) protocol, which is the first C1G2-complaint RFID data collection protocol. As an alternative to exact information collection, our GRC protocol is able to estimate the sensing data of un-sampled tags with a small set of collected data of the sampled tags. Second, we address some challenging issues when implementing GRC in RFID system, such as learning inherent data correlation, collecting data from sampled tags with Gen2 commands and estimating the un-sampled data with sampled data. Third, we conduct extensive simulations to evaluate the performance of the our GRC protocol. The results show that GRC takes only $1/28{\sim}1/3$ of the time compared with the state-of-the-art data collection schemes.

Acknowledgment

This work was supported in part by the State Key Program of National Natural Science of China under Grants 61432002 and 61832013, in part by the NSFC under Grants 61772251, 61772112, 61672379, U1701263 and 61702365 and in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS-1651947, and CNS 1564128.

References

[1] J. Liu, B. Xiao, K. Bu, and L. Chen, "Efficient distributed query processing in large RFID-enabled supply chains," in *Proc. of IEEE INFOCOM*, 2014.

[2] J. Yu, W. Gong, J. Liu, L. Chen, and K. Wang, "On efficient tree-based tag search in large-scale RFID systems," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 42–55, 2019.

[3] R. Zhang, H. Moungla, J. Yu, and A. Mehaoua, "Medium access for concurrent traffic in wireless body area networks: Protocol design and analysis," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2586–2599, 2017.

[4] J. Yu, L. Chen, R. Zhang, and K. Wang, "Finding needles in a haystack: Missing tag detection in large RFID systems," *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 2036–2047, 2017.

[5] X. Liu, J. Yin, S. Zhang, B. Ding, S. Guo, and K. Wang, "Range-based localization for sparse 3D sensor networks," *IEEE Internet of Things Journal*, vol. 27, no. 1, pp. 42–55, 2019.

[6] X. Liu, S. Zhang, B. Xiao, and K. Bu, "Flexible and time-efficient tag scanning with handheld readers," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 840–852, 2016.

[7] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu, "Anchor-free backscatter positioning for RFID tags with high accuracy," in *Proc. of IEEE INFOCOM*, 2014.

[8] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[9] K. Bu, B. Xiao, Q. Xiao, and S. Chen, "Efficient misplaced-tag pinpointing in large RFID systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2094–2106, 2012.

[10] X. Liu, K. Li, G. Min, K. Lin, B. Xiao, Y. Shen, and W. Qu, "Efficient unknown tag identification protocols in large-scale RFID systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3145–3155, 2014.

[11] C. Occhiuzzi, A. Rida, G. Marrocco, and M. Tentzeris, "RFID passive gas sensor integrating carbon nanotubes," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 10, pp. 2674–2684, 2011.

[12] D. J. Yeager, A. P. Sample, J. R. Smith, and J. R. Smith, "Wisp: A passively powered uhf RFID tag with sensing and computation," *RFID handbook: Applications, technology, security, and privacy*, pp. 261–278, 2008.

[13] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "Unknown-target information collection in sensor-enabled RFID systems," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1164–1175, 2014.

[14] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. of IEEE INFOCOM*, 2011.

[15] S. Madden, "Intel lab data," http://db.csail.mit.edu/labdata/labdata.html.

[16] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A time-efficient information collection protocol for large-scale RFID systems," in *Proc. of IEEE INFOCOM*, 2012.

[17] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient polling protocols in RFID systems," in *Proc. of ACM Mobihoc*, 2011.

[18] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *In Proc. of ICDE*, 2006.

[19] Z. Zhao, J. Bu, W. Dong, T. Gu, and X. Xu, "Coco+: Exploiting correlated core for energy efficient dissemination in wireless sensor networks," *Ad Hoc Networks*, vol. 37, pp. 404–417, 2016.

[20] Z. Zhao, W. Dong, J. Bu, Y. Gu, and C. Chen, "Link-correlation-aware data dissemination in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5747–5757, 2015.

[21] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *Springer European Workshop on Wireless Sensor Networks*, 2006.

[22] Z. Zhao, W. Dong, J. Bu, T. Gu, and G. Min, "Accurate and generic sender selection for bulk data dissemination in low-power wireless networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 948–959, 2017.

[23] X. Xie, X. Liu, W. Xue, K. Li, B. Xiao, and H. Qi, "Fast collection of data in sensor-augmented rfid networks," in *Proc. of IEEE SECON*, 2016.

[24] *EPC radio-frequency identity protocols class-1 gen-2 UHF RFID protocol for communications at 860MHz-960MHz, EPCglobal*, http://www.epcglobalinc.org/standards/uhfc1g2, 2011.

[25] L. Yang, Y. Chen, Xiang-Yang, C. Xiao, M. Li, and Y. Liu, "Tagoram:real-time tracking of mobile RFID tags to high precision using cots devices," in *Proc. of ACM MOBICOM*, 2014.

[26] A. Jedda, M. G. Khair, and H. T. Mouftah, "Distributed algorithms for the rfid coverage problem," in *Proc. of IEEE ICC*, 2013.

[27] Y. Gong, M. Shen, J. Zhang, O. Kaynak, W. Chen, and Z. Zhan, "Optimizing rfid network planning by using a particle swarm optimization algorithm with redundant reader elimination," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900–912, Nov 2012.

[28] L. M. Ni, D. Zhang, and M. R. Souryal, "RFID-based localization and tracking technologies," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 45–51, 2011.

[29] Q. Lin, L. Yang, Y. Sun, T. Liu, X. Li, and Y. Liu, "Beyond one-dollar mouse: A battery-free device for 3D human-computer interaction via RFID tags," in *Proc. of IEEE INFOCOM*, 2015.

[30] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. De Oliveira, R. B. De Araujo, and A. A. Loureiro, "An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks," *Elsevier Computer Communications*, vol. 36, no. 9, pp. 1054–1066, 2013.

[31] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing RFID tags in a time-efficient way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, 2015.

[32] W. Gong, I. Stojmenovic, A. Nayak, K. Liu, and H. Liu, "Fast and scalable counterfeits estimation for large-scale RFID systems," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1052–1064, 2016.

[33] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal, "Improved approximation algorithms for connected sensor cover," *Wireless Networks*, vol. 13, no. 2, pp. 153–164, 2007.

[34] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1399–1411, 2012.

[35] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. 6, pp. 235–284, 2008.

[36] A. Stetsko, L. Folkman, and V. Matyas, "Neighbor-based intrusion detection for wireless sensor networks," in *Proc. of IEEE ICWC*, 2010.

[37] X. Liu, X. Xie, K. Li, B. Xiao, J. Wu, H. Qi, and D. Lu, "Fast tracking the population of key tags in large-scale anonymous RFID systems," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 278–291, 2017.

[38] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. of The Second IEEE Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2005, pp. 166–172.

[39] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, "Efficient tag identification in mobile RFID systems," in *Proc. of IEEE INFOCOM*, 2010.

[40] H. Liu, W. Gong, X. Miao, K. Liu, and W. He, "Towards adaptive continuous scanning in large-scale RFID systems," in *Proc. of IEEE INFOCOM*, 2014.

[41] L. Pan and H. Wu, "Smart trend-traversal: a low delay and energy tag arbitration protocol for large RFID systems," in *Proc. of IEEE INFOCOM*, 2009.

[42] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for rfid identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 796–809, 2015.

[43] C. Liu, K. Wu, and J. Pei, “An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010–1023, July 2007.

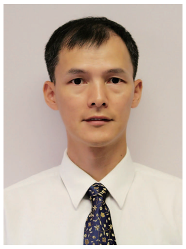
[44] Z. Zhao, W. Dong, G. Guan, J. Bu, T. Gu, and C. Chen, “Modeling link correlation in low-power wireless networks,” in *Proc. of IEEE INFOCOM*, 2015.

Xin Xie received the B.E. degree in computer science from Dalian University of Technology, Dalian, China, in 2013. He is currently pursuing the Ph.D. degree in Computer Science at Dalian University of Technology. He was a visiting scholar with the Department of Computer Sciences, Purdue University, USA, in 2018; His research interests include RFID and mobile sensing technologies.

Xiulong Liu is currently a postdoctoral fellow in School of Computing Science, Simon Fraser University, Canada. Before that, he received the B.E. degree and Ph.D. degree from the School of Software Technology and the School of Computer Science and Technology, Dalian University of Technology, China, in 2010 and 2016, respectively. He sequentially served as a visiting researcher in Aizu University (Japan) and a postdoctoral fellow in The Hong Kong Polytechnic University from Oct. 2015 to Mar. 2019. His research interests include RFID systems and wireless sensor networks. He has published more than 30 research papers in prestigious journals and conferences including TON, TMC, TC, TPDS, TCOM, INFOCOM, ICNP, *etc*. He received the Best Paper Awards from ICA3PP 2014 and IEEE System Journal 2017. He is also the recipient of CCF Outstanding Doctoral Dissertation award 2017.

Heng Qi is an associate professor at the School of Computer Science and Technology, Dalian University of Technology, China. He received bachelor’s degree from Hunan University in 2004 and master’s degree from Dalian University of Technology in 2006. Then he received his Ph.D. degree from Dalian University of Technology in 2012. His research interests include computer network, wireless network and multimedia computing.

Keqiu Li received the bachelor’s and master’s degrees from the Department of Applied Mathematics at the Dalian University of Technology in 1994 and 1997, respectively. He received the Ph.D. degree from the Graduate School of Information Science, Japan Advanced Institute of Science and Technology in 2005. He also has two-year postdoctoral experience in the University of Tokyo, Japan. He is currently a professor in the College of Intelligence and Computing, Tianjin University, China. He has published more than 100 technical papers, such as IEEE TPDS, ACM TOIT, and ACM TOMCCAP. He was an Associate Editor of IEEE TPDS and IEEE TC. His research interests include data center networks, cloud computing and wireless networks.



Bin Xiao received the B.Sc. and M.Sc. degrees in electronics engineering from Fudan University, China, in 1997 and 2000, respectively,and the Ph.D. degree in computer science from the University of Texas at Dallas in 2003. After his Ph.D. graduation, he joined the Hong Kong Polytechnic University as an assistant professor. Currently, he is an associate professor in the Department of Computing at The Hong Kong Polytechnic University, Hong Kong. His research interests include mobile cloud computing, data management, network security, wireless sensor networks, and RFID systems. He is an associate editor for the International Journal of Parallel, Emergent and Distributed Systems.

Jie Wu is the associate vice provost for international affairs with Temple University. He also serves as the chair and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Tempe University, he was a program director at the US National Science Foundation and was a distinguished professor with Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including the IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. He was general co-chair/chair of the IEEE Mobile Adhoc and Sensor Systems 2006, the IEEE International Parallel & Distributed Processing Symposium 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair of the IEEE Technical Committee on Distributed Processing (TCDP).