Thermotag: Item-level Temperature Sensing with a Passive RFID Tag

Xingyu Chen
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China

Jia Liu
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China

Fu Xiao
Department of Computer, Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu, China

Shigang Chen
Department of Computer & Information Science & Engineering, University of Florida
Gainesville, Florida, USA

Lijun Chen
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China

ABSTRACT

Temperature sensing plays a significant role in upholding quality assurance and meeting regulatory compliance in a wide variety of applications, such as fire safety and cold chain monitoring. However, existing temperature measurement devices are bulky, cost-prohibitive, or battery-powered, making item-level sensing and intelligence costly. In this paper, we present a novel tag-based thermometer called Thermotag, which uses a common passive RFID tag to sense the temperature with competitive advantages of being low-cost, battery-free, and robust to environmental conditions. The basic idea of Thermotag is that the resistance of a semiconductor diode in a tag's chip is temperature-sensitive. By measuring the discharging period through the reverse-polarized diode, we can estimate the temperature indirectly. We propose a standards-compliant measurement scheme of the discharging period by using a tag's volatile memory and build a mapping model between the discharging period and temperature for accurate and reliable temperature sensing. We implement Thermotag using a commercial off-the-shelf RFID system, with no need for any firmware or hardware modifications. Extensive experiments show that the temperature measurement has a large span ranging from 0 °C to 85 °C and a mean error of 2.7 °C.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**.

KEYWORDS

Passive RFID, Temperature Sensing, Persistence Time

ACM Reference Format:
Xingyu Chen, Jia Liu, Fu Xiao, Shigang Chen, and Lijun Chen. 2021. Thermotag: Item-level Temperature Sensing with a Passive RFID Tag. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24-July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3458864.3467879

MobiSys '21, June 24-July 2, 2021, Virtual, WI, USA

ACM ISBN 978-1-4503-8443-8/21/06…$15.00
https://doi.org/10.1145/3458864.3467879

1 INTRODUCTION

Temperature is a physical quantity that presents hot and cold. The properties of materials and almost all physical, chemical, and biological processes are temperature dependent. Sensing temperature plays a significant role in maintaining operational efficiency, upholding quality assurance, and meeting regulatory compliance in a variety of businesses, such as fire safety [15], warehouse management [12, 38], pharmaceutical industries [34], and greenhouses [18]. For example, temperature sensors can provide an early alarm of fire accidents and help save properties and human lives. In the agriculture industry, greenhouses need to monitor temperature within a resolution of around 4 °C for benefiting plant growth [19]. In cold chain, food supplies or pharmaceutical companies need to monitor and document temperatures of temperature-sensitive goods throughout processing, transportation, and distribution for preventing perishable food contamination or maintaining the efficacy of certain drugs. In data centers, excessive heat can quickly bring a fully functional computer room to a halt. Temperature monitors help meet the demand for constant and uninterrupted uptime.

Nowadays, there are many types of temperature sensors with different features. The core property of these sensors is that they have some parameters that vary linearly with the temperature; any change of the temperature gives rise to variations in certain parameters. For instance, glass thermometers detect temperature through the volume of the liquid in a glass tube [4, 16, 20], resistance thermometers obtain temperature by checking the resistance of a specific material via

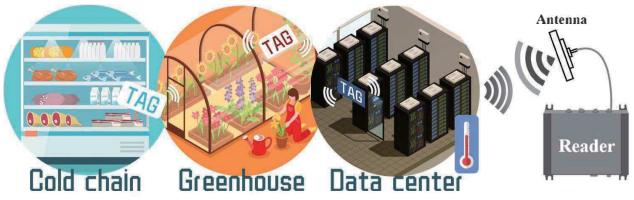

Figure 1: Item-level temperature sensing with Thermotag.

an accurate resistance-temperature relationship [7, 22], thermocouple measures temperature by observing the voltage values developed by the temperature related electronic effect [13, 23], and infrared thermometers capture temperature from a portion of the thermal radiation emitted by the object [2, 17, 28]. In spite of the wide use of these sensors, they suffer from three limitations. First, most existing sensors are expensive, making item-level temperature monitoring cost-prohibitive. Second, most existing sensors require external power to activate the internal electronic components, making the sensor system bulky and requiring high installation or replacement costs. Third, they cannot achieve wireless temperature sensing directly; an extra embedded circuit (e.g., RF transmitter) is required to perform data transmission [18, 38].

Given that commodity radio frequency identification (RFID) tags are already widely used in a number of applications for good tracking and management [11, 14, 26, 29, 32, 35, 36], RFID is a good choice for temperature sensing due to its compelling features of being low-cost, small-sized, and battery-free [10, 24, 25, 27, 30, 33]. RFID-based temperature sensing generally falls into two categories. The first is to combine a tag with a traditional temperature sensor, in which the sensor is in charge of temperature measurement and the tag is responsible for data transmission over the air [5, 6, 10, 24, 25, 27, 30]. Although the sensor-augmented tag is feasible, adding an extra temperature sensor does not reduce the cost. In addition, since the temperature sensor largely consumes the energy captured by the tag from electromagnetic waves, the communication range between the reader and the tag declines sharply, which imposes restrictions on the applicability of RFID [18, 38].

To address this problem, recent work [33] attempts to sense temperature with commodity RFID tags. The basic idea is that RF signals modulated by a tag vary with the temperature. By observing the fluctuations of received RF phase values, the proposed approach can estimate the temperature level approximately. However, the biggest problem is that the RF parameter is strongly dependent on the environmental conditions and signal acquisition factors, such as tag position, tag orientation, communication power, and multi-path effects. Two different measurements of the same temperature are very likely to produce two totally different RF parameters. RTSense [19] tries to handle this problem by using a pair of tags and the phase difference of the two tags can eliminate the impact of environmental factors. This works to some extent but needs each tag to carry a custom-designed antenna, which cannot be deployed in commodity RFID systems directly.

In this paper, we design and implement Thermotag, a novel temperature sensing system with commodity RFID systems. Thermotag takes the internal electrical components of the tag's circuit itself as the vehicle to measure the temperature, which is robust to various environmental conditions. Besides, since Thermotag uses common commercial passive tags as is, the temperature sensing system is cost-effective (0.02-0.03 USD per tag) and long-lasting (battery-free), resulting in practical item-level temperature monitoring. The basic idea of Thermotag is that high temperature increases the resistance of the diode in a tag's chip, as the atoms have more kinetic energy and electrons (current) encounters greater flowing difficulty in this case. By measuring the current variances when the temperature changes, it is promising to implement a tag-based thermometer. To obtain the current variances, an intuitive solution is to set up an electronic test circuit and monitor the current manually. This concept works in theory but suffers from three problems in practice. First, the size of the tag chip is less than 1 mm, which is too small to be measured. Second, performing measurements individually and manually is extremely labor intensive and time consuming, especially when many tagged objects must be monitored. Third, a purpose-built electronic test platform greatly increases the cost of temperature sensing, which is neither practical nor scalable in commercial use.

To address these problems, we resort to an indicator called persistence time, which can reflect the current variances around the tag circuit indirectly. We notice that for proper functioning, the passive RFID tag needs a capacitor to store the energy captured from RF signals for data storage in volatile memory. Once the capacitor is fully charged, if we remove the voltage source (e.g., turn off the reader), the circuit starts discharging. The time span from the initial supply voltage decay to the voltage threshold when the volatile data are lost is referred to as *persistence time*. Clearly, persistence time presents how fast the discharging process is and further indirectly reflects current variances of the tag circuit. We can sense the temperature accordingly by measuring the persistence time and building the mapping relationship between the persistence time and the temperature. The main contributions are three-fold.

- We present a temperature sensing system called Thermotag that takes a common passive tag as the thermometer by observing current variances around the tag circuit. The competitive advantages of Thermotag are battery-free, low-cost, robust to environmental conditions, and fully compatible with the existing RFID standard, thus making item-level temperature monitoring possible.
- We use persistence time to indirectly reflect current variances around the tag circuit. A standards-compliant scheme is designed to measure the persistence time by observing the flag bit of the tag's volatile memory. Then a mapping model between the persistence time and temperature is built for fine-grained temperature sensing.

• We implement Thermotag using a commercial off-the-shelf RFID system, with no need for any hardware modifications. Extensive experiments show that Thermotag has a large measurement span from 0 °C to 85 °C and a mean measurement error of 2.7 °C.

The rest of this paper is organized as follows. Section 2 presents the system deployment and basic idea. Section 3 details the system model of Thermotag. Section 4 shows the extraction of persistence time. Section 5 discusses the temperature estimation. Section 6 evaluates the performance of Thermotag. Section 7 introduces the related work. Finally, Section 8 concludes this work.

2 SYSTEM SETUP AND BASIC IDEA

RFID exploits electromagnetic fields to identify and track tagged objects in a non-contact way. As shown in Fig. 1, a basic RFID system typically consists of three components: a tag, a reader, and an antenna. The reader communicates with the tag through the antenna, which is the first point of contact for tag reading. Each RFID tag has a unique digital identity to label tagged objects and brings item intelligence in the era of Internet of Things (IoT). As the (passive) tag does not have a built-in power source, it gathers energy from the reader antenna's electromagnetic RF signals, which makes it small, cost-effective, long-lasting, and thus popular in a wide variety of applications.

In this paper, we design and implement Thermotag, a lightweight tag-based thermometer that measures the temperature of a target through a passive RFID tag, with no need for any hardware modifications. Fig. 1 shows the system setup of Thermotag. An unmodified commodity RFID tag is attached to a target, which could be fresh milk in cold chain, plants in greenhouses, or a functional server in a data center. By measuring the temperature of the tag, we can sense the tagged person or object in real time. However, as the tag does not have an independent temperature sensor, determining the temperature directly is not easy.

The basic idea of Thermotag is to find a temperature-sensitive electrical component of a tag circuit and build the mapping relationship between the electrical component and the temperature. Taking a closer look at a tag, we can see that it generally consists of two components: a tag antenna and a tag chip. The tag chip is small but complete and has many semiconductor devices and diodes. Temperature has great impact on diodes: when temperature rises, the internal resistance of a diode rises because the atoms have more kinetic energy and electrons (current) experience greater flow difficulty. If we can measure the current variances as the temperature goes up or down, it is promising to implement a tag-based thermometer. An intuitive solution is to set up an electronic test circuit and monitor the current manually. This concept works in theory but suffers from three problems in practice. First, the size of the tag chip is less than 1 mm, which is too small to be measured. Second, performing measurements individually and manually is extremely labor intensive and time consuming, especially when many tagged objects need to be monitored concurrently. Third, a purpose-built electronic test platform greatly increases the cost of temperature sensing, which is neither practical nor scalable in commercial use.

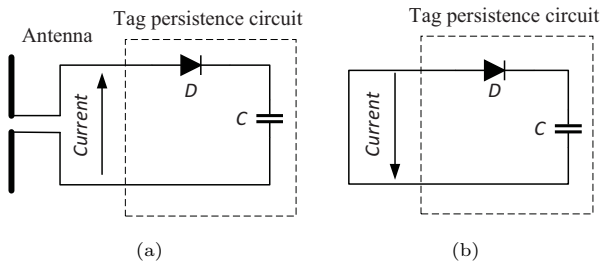

Figure 2: Persistence circuit: (a) Charging; (b) Discharging.

To address this problem, we resort to persistence time, which can indirectly reflect the current variances around the tag circuit. More specifically, a tag chip has at least one persistence circuit of volatile data for proper functioning, which is specified by UHF RFID standard EPCglobal Gen2 (Gen2) [1]. As shown in Fig. 2, the persistence circuit has two components: a capacitor and a diode. When RF signals are received by the tag antenna, the voltage generated on the tag antenna powers up the capacitor. Once the capacitor is fully charged, the circuit starts discharging and the current leaks through the reverse-polarized semiconductor diode if we remove the voltage source (turn off the reader for example). We refer to the time span from the initial supply voltage decay to the voltage threshold when the volatile data are lost as *persistence time*. Clearly, the persistence time presents how fast the discharging process is and further indirectly reflects the speed of current leakage through the reverse-polarized semiconductor diode. Notably, in order to provide the tag chip with a stable voltage, a commercial tag is required to carry a low dropout regulator [39], which uses a voltage reference block to produce a regulated and constant voltage when the capacitor is fully charged, regardless of the communication distance and the transmit power of the reader. Thus, persistence time is a good indicator of temperature.

Thermotag has three competitive advantages. First, it is fully compatible with the existing RFID standard and plug-n-play, with no need for any modifications of hardware. Second, the tag-based thermometer costs only about two cents, which is much cheaper than existing temperature sensors and has the potential to yield large savings in this industry. Third, Thermotag is more robust to environmental conditions, such as tag position, communication distance, and multi-path effects than RF signals, making it a good choice for practical use.

3 PERSISTENCE TIME AND TEMPERATURE

In this section, we study the relationship between persistence time and the temperature. Fig. 3 shows a persistence circuit for a tag's volatile memory, which contains four parts: a

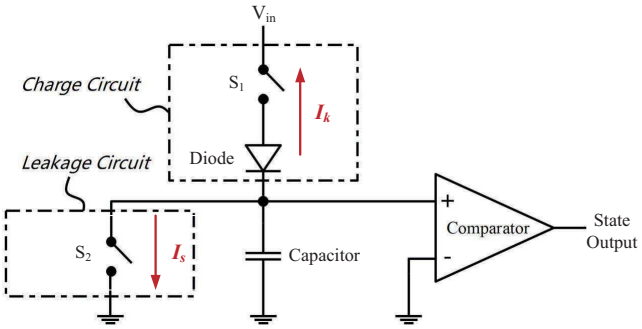

Figure 3: Persistence circuit.

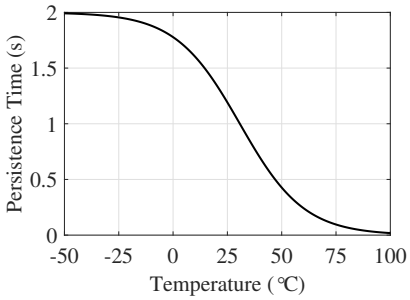

Figure 4: The theoretical relationship between persistence time and temperature.

charging circuit (the switch S_1 and the semiconductor diode), a leakage circuit (the switch S_2), a capacitor, and a voltage comparator [8]. As aforementioned, a passive tag does not have a built-in power source, which can capture electric energy from the reader antenna's electromagnetic waves. When the switch S_1 is on and the switch S_2 is opened, the high-input voltage V_{in} will charge up the capacitor through the diode. Once the capacitor is fully charged, the circuit starts discharging if we remove the power resource. The persistence of the stored energy in the capacitor is determined by the semiconductor leakage: mobile charges move through the reverse-biased diode and switches until its voltage decays to 0. The discharging current of the tag persistence circuit consists of two parts. The first part is the leakage current that flows through the diode and the switch S_1, which is denoted as I_k in Fig. 3. In theory, a reverse-biased diode should not conduct any current. However, due to an increased barrier potential, the free electrons on the p side are dragged to the battery's positive terminal, producing a current of minority charge carriers. This leakage current is temperature-sensitive and a common rule of thumb is that it doubles for every c degree Celsius rise in temperature [3]. Hence, the leakage current can be expressed as $I_k = I_0 \sqrt[c]{2}^t$, where I_0 is the reference current at 0 °C and t represents the temperature in degree Celsius. The second part is the supplement leakage current through the switch S_2, denoted as I_s. Unlike I_k, as no diode exists, I_s remains stable as the temperature fluctuates. This persistence circuit provides a tag with a specific persistence time that is specified by the RFID standard. We have the persistence time $T(t)$ under the temperature t:

$$T(t) = \frac{Q}{I_k + I_s} = \frac{CV_{in}}{I_0 \sqrt[c]{2}^t + I_s}, \tag{1}$$

where $Q = CV_{in}$ represents the electronic charging held by the capacitor C. By a series of formula transforms from Eq. (1), we have:

$$T(t) = \frac{a}{\sqrt[c]{2}^t + b}, \tag{2}$$

where $a = \frac{CV_{in}}{I_0}$ and $b = \frac{I_s}{I_0}$. The terms a, b, and c are three constants determined by a tag's electronic components. Fig. 4 plots the theoretical pattern of persistence time as the temperature varies from -50 °C to 100 °C. As can be seen, persistence time declines monotonously and each plot exclusively corresponds to a temperature. With this mapping model, we can estimate the temperature once a specific persistence time is given. Next, we detail the process of extracting persistence time with commercial RFID devices.

4 MEASUREMENT OF PERSISTENCE TIME

The basic idea of measuring persistence time is to build a fully charged RC circuit and then run the discharging operation. This approach requires the following three steps. As shown in Fig. 5, first, we turn on the reader and let it issue the RF signals to energize the tag. Second, once the tag is fully charged, we cut off the power by turning off the reader; the discharging process thus begins. Third, after a period of time T_p, we check whether the tag is exhausted or not. By gradually increasing the time period T_p and repeating the above three steps, we can find a maximum of T_p that is guaranteed to help the tag work properly. This maximum is actually the persistence time to be measured.

Among the three steps mentioned above, the first two, turning the reader on and off, are easy to operate. However, examining when the power of the tag is exhausted with a commodity RFID system is a challenging task. To address this problem, we resort to the volatile memory of a tag. Unlike non-volatile memory (e.g., NAND flash and solid-state drives), the volatile memory requires power to maintain the stored information. Once the power is cut off (or lower than a threshold), the stored data are quickly lost. In the RFID standard Gen2, we find a metric *inventoried flag*, which is a one-bit indicator in a tag's volatile memory. By flipping the inventoried flag and checking its status continuously, we are able to know when the power of the tag is exhausted. Next, we first introduce the Gen2 protocol and then detail the process of measuring persistence time based on the RFID standard.

4.1 EPCglobal Gen2 Protocol

The EPCglobal Gen2 (Gen2) protocol is a worldwide UHF RFID standard that defines the physical interactions and

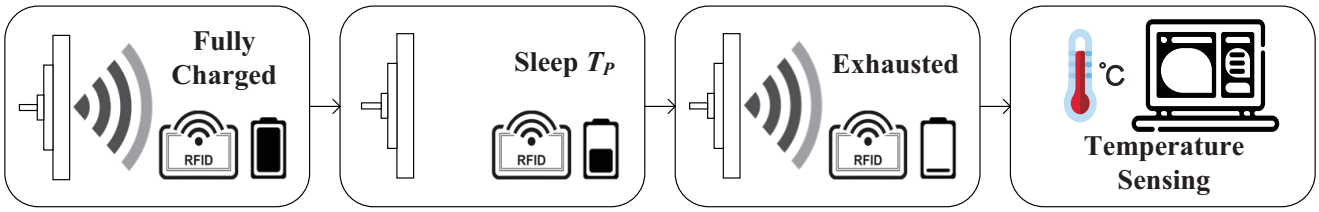

Figure 5: Basic idea of feature extraction.

logical operating procedures between the readers and tags [1]. On the basis of Gen2, we highlight the related functions that will be involved by Thermotag below.

Tag Memory. Gen2 standard specifies that the tag memory is supposed to contain four distinct memory banks (page 44—51 in [1]). MemBank-0 is reserved for kill and access passwords if encryption is implemented on the tag. MemBank-1 stores the electronic product code (EPC), i.e., tag ID that is often referred to. MemBank-2 stores TID that indicates the tag- and manufacturer-specific data at the time of manufacture, which is permalocked and unchangeable. MemBank-3 is user memory that allows customized data storage. In this work, we need to visit the tag's ID, so MemBank-1 is used.

Sessions & Inventoried Flags. Gen2 requires the readers and tags to provide four sessions (denoted as S0, S1, S2, and S3). Tags in one of these sessions shall neither use nor modify an inventoried flag for a different session. This allows two or more readers to use different sessions to independently inventory a common tag population (in different time slots). The inventoried flag is actually a one-bit indicator of a tag's volatile memory. The binary state of the inventoried flag is denoted by A and B, respectively, where A is the initial state as usual. The volatile memory requires power to maintain the stored information. Once the power is lower than a threshold, the stored data are quickly lost, that is, the inventoried flag will flip to A when the power of the tag is exhausted, no matter what the previous state is. Next, we take the inventoried flag in S1 as an example to show how our system works; the other sessions can be used in the same way.

Select. *Select* is a mandatory command that is prior to each inventory round. It allows a reader to choose a specific subset of tags that participate in the subsequent inventoried round. Aside from tag selection, the *Select* command can also assert or deassert a tag's selected (SL) flag, or set a tag's inventoried flag to either A or B. These flags are used to determine whether or not a tag may respond to a reader. Specifically, a *Select* command consists six fields.

Table 1: Eight Actions of *Select*.

Action	Tag Matching	Tag Not-Matching	Abbr.
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*	AB
001	assert **SL** or **inventoried** → *A*	do nothing	A-
010	do nothing	deassert **SL** or **inventoried** → *B*	-B
011	negate **SL** or (*A*→*B*, *B*→*A*)	do nothing	S-
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*	BA
101	deassert **SL** or **inventoried** → *B*	do nothing	B-
110	do nothing	assert **SL** or **inventoried** → *A*	-A
111	do nothing	negate **SL** or (*A*→*B*, *B*→*A*)	-S

- *MemBank, Mask, Length, Pointer.* These four fields jointly determine which tags are matching or not. *MemBank* specifies which memory bank is chosen for comparison. As aforementioned, four memory banks are available, MemBank-0, MemBank-1, MemBank-2, and MemBank-3, which are indicated by 0, 1, 2, and 3, respectively. *Pointer* indicates the starting position in the chosen memory bank. *Length* determines the length of *Mask*, which is a customized bit string according to the user demands. If *Mask* is the same as the string that begins at *Pointer* and ends *Length* bits later in the memory of *MemBank*, then the corresponding tag is matched.
- *Target, Action.* The field *Target* indicates the object that *Select* will operate, which is either a tag's SL flag or an inventoried flag in any one of four sessions. The sessions are specified by the Gen2 protocol to fit the case of exclusive reading among multiple readers. Therefore, five different targets can be chosen. The selection function is actually achieved by masking the interested tags, setting the matching tags' inventoried flags or SL flag to a specific state while not-matching tags to opposite, and finally operating the tags with the same flag state. How to set the inventoried flag and the SL fag is determined by the *Action* field. As shown in Table 1, eight actions are available, where matching and not-matching tags set their inventoried flags to A or B. By combining *Target* and *Action*, the reader is able to modify the state of the inventoried flags or the SL flag for a group of tags. For example, when the *Action* is 0, the matching tags are set to A while the not-matching tags are set to B. The term "do nothing" means the tags keep their flags unchanged.

Query. *Query* command starts a new inventory round over the tag subpopulation, which are chosen by the previous *Select* command(s). In the inventory round, the reader will carry out a frame that consists of some time slots. Each "selected" tag randomly picks one of these time slots and transmits its tag ID to the reader in that slot. After a tag is queried by the reader, it will invert its inventoried flag, i.e., from the state A to B, or vice versa. *Query* includes three fields that we would like to focus on.

• *Session, Target*. Similar to that in *Select*, this field *Session* in *Query* specifies one of the four sessions used in the incoming inventory round. The field $Target$ determines which tags will participate in the current inventory round, where 0 indicates the tags with the inventoried flag being A and 1 indicates B.

• *Sel*. This field consists of two bits that determine which tags respond to *Query*: 00_2 and 01_2 indicate all matching tags in the previous Select command; 10_2 indicates tags with deasserted SL flag ($\sim SL$); and 11_2 indicates tags with asserted SL flag (SL).

On the basis of the above Gen2-compatible functions, we next detail how to jointly utilize the *Select* and *Query* commands to measure the persistence time by using the state of the inventoried flag. The method is called flag-based measurement.

4.2 Flag-based Measurement (FM)

This technique follows a basic idea: when the internal energy of a tag is exhausted, the inventoried flag moves back to the initial state A, regardless of its previous state. If we set the tag's inventoried flag to B and keep the RC circuit fully charged, then the time period from starting discharging to the time when the inventoried flag turns to A can be treated as the persistence time.

To measure the discharging time, we need to jointly use the *Select* command and the *Query* command. According to Gen2, a *Select* command can be written as follows:

$$S(\underbrace{t}_{\text{Target}}, \overbrace{a}^{\text{Action}}, \underbrace{b}_{\text{Membank}}, \overbrace{p}^{\text{Pointer}}, \underbrace{l}_{\text{Length}}, \overbrace{k}^{\text{Mask}}). \tag{3}$$

To set a tag's inventoried flag to B, the reader just needs to broadcast a *Select* as follows:

$$\text{Flag} \leftarrow \text{BA: } S(1, 4, 1, 32, 96, id), \tag{4}$$

where $t = 1$ (001_2) means the operating object is set to the inventoried flag in session 1 (S1), $a = 4$ indicates that the inventoried flags of matching tags will be set to B, while those of not-matching tags will be set to A, $(b, p, l, k) = (1, 32, 96, id)$ means the tag's ID is the same as id is selected (matching). Note that the first bit of the tag ID starts from the 32nd bit ($p = 32$) in MemBank-1, because the first 32 bits are a protocol-control (PC) word and the tag ID follows behind the PC word. More details can be seen in [1].

Following this process, the target tag is set to B. Now the question is how long we should wait before we can obtain a fully charged RC circuit. Gen2 specifies that the charging time should be no longer than 2 ms, which is much less than the time period (about 20 ms) for broadcasting a select command. In other words, once the select command in (4) is carried out, the target tag has the inventoried flag being B as well as the RC circuit being fully charged.

Afterwards, we move to the discharging process by turning off the readers. Given that the tag can no longer harvest energy from the reader, the stored electric energy is consumed gradually. After a period of time T_p for discharging, the reader broadcasts a query command to check whether any tag with the inventoried flag B exists. The query command is as follows:

$$\text{Query } B: \ Q(Session = 1, Target = 1, Sel = 0). \tag{5}$$

If a tag reply is received, it means that the persistence time of this tag is longer than T_p. In this case, we need to increase T_p by a small step Δt and repeat the above select-query process again. For the first time period T_p that makes no tag reply, it is treated as the persistence time to be measured. This is because no tag reply means that the tag's inventoried flag has flipped to A as the power ran out.

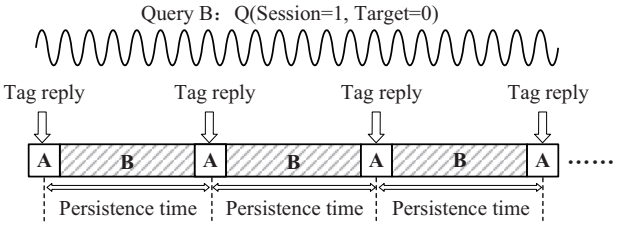

Figure 6: Enhanced FM for obtaining persistence time.

4.3 Enhanced FM

With above method, we are able to extract persistence time from a tag. However, a large gap still exists between FM and efficient temperature sensing because that the process of increasingly adjusting the waiting time T_p is time-consuming. For example, assume a tag's persistence time is 3 s and the step length is 0.1 s. The waiting time T_p is initialized to 0.1 s and FM needs to iteratively try 0.1, 0.2, 0.3, ..., 3.0 s. Summing up the overhead of each attempt, we have the overall time cost 46.5 s. This time cost is appropriate for applications without real-time requirement. However, in applications like a fire-alarm system, this time may be too long to be considered applicable for practical use.

The basic reason for the low time efficiency is that when a waiting time T_p is examined, we need to reset all tags and retry the next one. A longer time is needed for checking. If we can run the measurement within just one waiting time window, the performance will be improved greatly. Through extensive experiments, we find that the query command does not charge the tag in the session S1. In other words, during the discharging process, we are able to keep querying the tags, with no need to turn off the reader. Once a tag is queried by the reader, it will be recharged again.

With these features, the enhanced FM measures the tag t's persistence time as follows. First, similar to the basic FM, the reader broadcasts a select command (see Eq. (4)) with action BA to set t's inventoried flag to B. After that, the discharging process starts and the reader queries the tag with the flag state being A. The query command is:

$$\text{Query } A: \ Q(Session = 1, Target = 0). \tag{6}$$

As shown in Fig. 6, during the discharging process, the internal circuit energizes the tag and keeps the inventoried flag B, so the reader cannot receive any response from the tag t. When the power level is too low to maintain the information of the volatile memory, the inventoried flag moves back to the initial state A. At that time, because the reader keeps querying tags with A, the tag t satisfying this condition will reply to the reader. By observing the time span from the start of discharging to the tag reply, we are able to derive the persistence time of the tag. Clearly, the enhanced FM does not need to try different waiting times; only one time window is sufficient to measure persistence time, thus saving a great number of overheads. For example, consider the above tag with 3 s persistence time. Enhanced FM results in great performance improvement, reducing the time from 46.5 s to only 3 s, compared to the basic FM.

After responding to the reader, the tag flips its inventoried flag to B (according to Gen2); meanwhile, the RC circuit is fully charged. With the reader continuing to query A, the tag will reply after another persistence time passes. Hence, if we need multiple measures of persistence time, we just need to record each time interval between two adjacent tag responses, as shown in Fig. 6. In fact, we can also simplify the enhanced FM by removing the select command, that is, the reader directly enters the inventory stage. By keeping querying tags with A, the reader can obtain each tag's replies. The time interval between any two adjacent tag replies is the tag's persistence time.

4.4 Dense Tag

So far, we have discussed how can we use enhanced FM to measure the persistence time from a tag. This can be easily generalized to multi-tag case by measuring the time interval between each tag's two adjacent replies. When the number of tags is small (i.e., 10 tags), this method functions well. However, as the number of tags increases, a congestion problem arises: a tag cannot be queried by the reader immediately as many tags with flag A wait in line to reply to the reader, thus resulting in measurement error of persistence time. To address this problem, we first divide a large tag set into several small subsets (each of them consisting of about 10 tags at most) and then run enhanced FM on each tag subset exclusively.

We assume that the tag set is τ and $\tau' \subseteq \tau$ is one of subsets to be measured. Intuitively, we can isolate all tags in τ' from $\tau - \tau'$ with the inventoried flag. Although this process works, the problem is that when we set the inventoried flags of τ' to B with the select command, the tags in $\tau - \tau'$ will be set to A. In the follow-up inventory stage, the reader queries tags with flag A, and these tags $\tau - \tau'$ will attend to respond. As a result, the tags in τ' fail to give a prompt reply when their flags move back to A due to power down. Setting $\tau - \tau'$ to B initially does not work either. To address this dilemma, we resort to another indicator: the SL flag. It has two states, which are denoted by SL and $\sim SL$, respectively. The reader can specify a set of tags in one of the two states

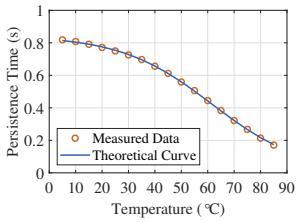

Figure 7: Curve fitting of measured data.

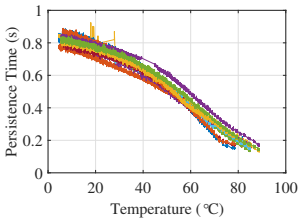

Figure 8: Persistence times extracted from 10 tags.

that will participate in the inventory round. The SL flag and the inventoried flag are independent and can be jointly used to remove the interference of $\tau - \tau'$. The solution is to set the target tags τ' to SL while other $\tau - \tau'$ are set to $\sim SL$. In the inventory stage, we let only the tags with SL participate in the response. By this means, even if the tag in $\tau - \tau'$ is with the inventoried flag A, it has to keep silent to the command of querying A. Specifically, we assume that $\tau' = \{t_1, t_2, ..., t_m\}$. The reader broadcasts the select commands as follows.

$$\begin{aligned} &① \ t_1 \leftarrow AB : S(t = 4, 4, 1, 32, 96, id_1), \\ &② \ t_i \leftarrow A- : S(t = 4, 5, 1, 32, 96, id_i), \ i \in [2, m], \end{aligned} \tag{7}$$

where the $Target$ field being set to 4 ($t = 4$) means that the operating object of the select is the SL flag. With above select commands, the SL flags of the tags in τ' are asserted (SL) while those of tags in $\tau - \tau'$ are deasserted ($\sim SL$). After that, we move to the inventory stage with the following query command:

$$\text{Query } A \ \& \ SL : \ Q(1, Target = 0, Sel = 3), \tag{8}$$

where the fields $Sel = 3$ and $Target = 0$ mean that the reader queries only the tags with the inventoried flags being A together with asserted SL. In such a context, only the target tags of τ' have the chance to reply; other tags in $\tau - \tau'$ are silenced due to $\sim SL$. For any target tag, we can obtain its persistence time by recording the time interval between two adjacent replies.

5 TEMPERATURE ESTIMATION

In this section, we detail the process of sensing the temperature with the measured persistence time. Consider Eq. (2) that depicts the theoretical relationship between the persistence time $T(t)$ and the temperature t, i.e., $T(t) = \frac{a}{\sqrt[c]{2^t} + b}$, where a, b, and c are three constants. Intuitively, for a specific tag, we can derive three constants a, b, and c first and then use the theoretical curve to predict the temperature once a specific persistence time is given. A common way to obtain a, b, and c is to use the least squares method. Specifically, we first extract n persistence time readings $\{(T_1, t_1), (T_2, t_2), \ldots, (T_n, t_n)\}$, where (T_i, t_i) represents the i-th measured persistence time at the temperature t_i, $1 \leq i \leq n$. Then, we utilize the least squares method to

derive a, b, and c as follows:

$$\min_{a,b,c} \sum_{i=1}^{n} ||T_i - \frac{a}{\sqrt[c]{2}^{t_i} + b}||^2. \quad (9)$$

The best fit for the set of persistence time $\{(T_i, t_i)\}$ happens when a, b, and c jointly minimize the sum of the offsets of the measured data points from the theoretical curve. As shown in Fig. 7, we use a real experiment to illustrate this process. We collect the persistence time from a passive tag at different temperature levels, ranging from 5 °C to 85 °C at a step of 5 °C. Clearly, the tightness between the theoretical plots and the measured values well indicates that the correctness of the time-temperature mapping model in Eq. (2). When a, b, and c are known, the temperature can be estimated by a series of formula transforms from Eq. (2):

$$t = \frac{\ln(\frac{a}{T} - b)}{\ln(\sqrt[c]{2})}, \quad (10)$$

where the temperature t is a monotonic function of the persistence time T. Given a persistence time, the temperature can be easily computed as desired. Hence, we need to do per-tag calibration for collecting the set $\{(T_i, t_i)\}$ and then obtain the best fit of (a, b, c) before it is put into use. This process, however, might suffer from a long delay, especially in the multi-tag case. To reduce overhead, we can calibrate multiple tags in parallel, as shown in Section 4.4. To further simplify this process, we find through extensive experiments that the tags with the same model have similar curve patterns and parameters (a, b, c). As shown in Fig. 8, we plot the set $\{(T_i, t_i)\}$ of persistence time extracted from 10 different passive tags with the same model. Clearly, although the tag diversity has some impact on persistence time, the curves of different tags experience a similar declining trend as temperature increases. In this way, given a tag model, we can collect the set $\{(T_i, t_i)\}$ from some tags in advance. After that, we derive the best fits with multiple (a, b, c)s. The mean of these (a, b, c)s is treated as the fitting parameters of this tag model. We refer to this approximate solution as one-shot calibration, which is a trade-off between the calibration overhead and accuracy. We will evaluate one-shot calibration in in Section 6.2.

6 EVALUATION

6.1 System Setup and Implementation

We build a prototype of Thermotag using commercial RFID readers and tags, with no need for any hardware or firmware modifications. As shown in Fig. 9, an Impinj H47 tag [9] is attached to a water-filled cylindrical glass container (with a radius of 3.75 cm and a height of 9.5 cm) to sense the water temperature. The reason for choosing water as the target is that it can be easily heated and also has 4.23 times more specific heat capacity than air, thus making it easier to keep the same temperature for a period of time. The tag is interrogated by an Impinj R420 reader equipped with a Larid S9028 antenna. An electronic stove is placed under the container for heating the water. A PT100 temperature sensor is placed into the container and is connected to an industrial online monitoring analyzer from Sinomeasure to provide the ground truth of the temperature of water in real time. A laptop with a 2.4 GHz CPU (Intel i5-6200U) and 8 GB memory is used to obtain the persistence time measured by the tag-based Thermotag together with the corresponding temperature level collected by the PT100. The temperature varies from 0 °C to 85 °C, at a step of 5 °C. Notice that this temperature range is determined by our system setup, which actually can be as large as [−40 °C, 85 °C], which is the tag's operating temperature specified by the datasheet [9]. Each plot in the experiments is the output from 100 tags on average.

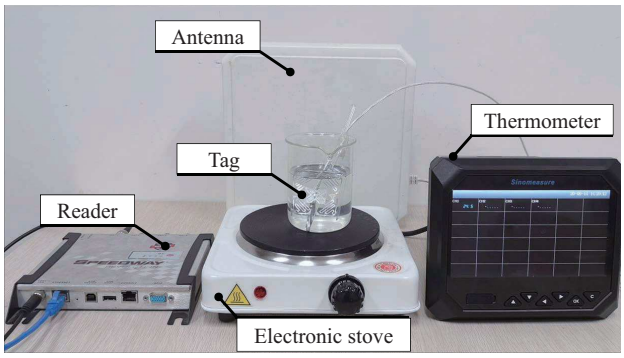

Figure 9: System deployment.

6.2 Accuracy

We first study the accuracy of per-tag calibration and one-shot calibration. As shown in Fig. 10, per-tag calibration has a higher accuracy than one-shot calibration over different temperatures, which has a mean error of 2.7 °C. This is intuitive, because per-tag calibration takes the tag diversity into account. The accuracy of one-shot calibration increases with temperature. The reason is that persistence time declines slowly at first and then more sharply as temperature increases from 0 °C to 85 °C, as shown in Fig. 8. The same error caused by diversity approximation in this case will result in a larger temperature error when the temperature is low. When temperature is over 30 °C, one-shot calibration has a small mean error of 3.0 °C, which is close to per-tag calibration. It is a trade-off between the accuracy and the calibration overhead, which is determined by specific applications. In what follows, we use per-tag calibration by default for the performance evaluation.

Next we compare the measurement accuracy of Thermotag with existing advanced work, including DMRT [31], RFThermometer [33], and RTSense [19]. As shown in Table 2, DMRT takes the differential minimum threshold power as the metric to perform the temperature measurement, with an error beyond 10 °C due to the low resolution of the metric. RFThermometer makes a fresh attempt to sense the temperature by using the RF phase that is sensitive to temperature changes. However, despite its high accuracy, the RF phase

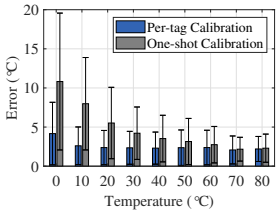

Figure 10: Accuracy comparison.

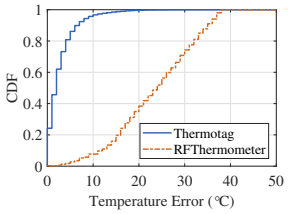

Figure 11: CDF of errors.

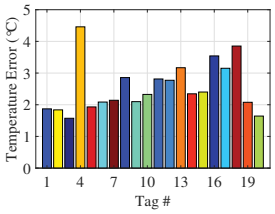

Figure 12: Tag diversity.

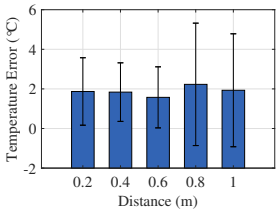

Figure 13: Impact of tag distance.

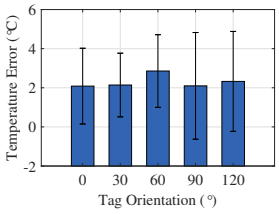

Figure 14: Impact of tag orientation.

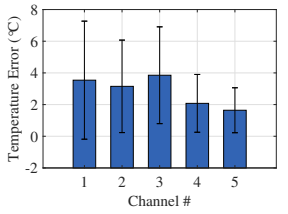

Figure 15: Impact of channel.

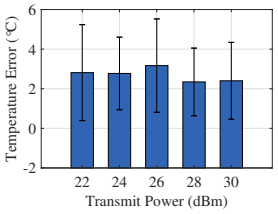

Figure 16: Impact of transmit power.

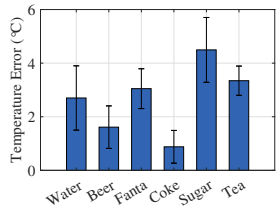

Figure 17: Impact of target material.

strongly depends on environmental factors, such as the tag location, which is not scalable to most practical scenarios. The most recent work RTSense uses a pair of tags to eliminate the negative impact caused by environmental factors and achieves a high accuracy of 2.9 °C. However, RTSense requires each tag to carry a custom-designed antenna and thereby cannot be applied to existing commodity RFID tags. Thermotag performs the best with a mean error of 2.7 °C. More importantly, Thermotag is very robust to environmental factors and can be deployed on commercial RFID tags directly, with no need for any hardware modifications.

Fig. 11 further presents the CDFs of sensing errors through Themotag and RFThermometer [33] that can function properly on commercially available tags. Clearly, our method Thermotag is far superior to RFThermometer. The 80th percentile of errors of Thermotag is less than 4 °C and the mean of the errors is only 2.7 °C. For RFThermometer, however, the 80th percentile of errors is over 30 °C and the mean of the errors is over 20 °C. The large performance gap is caused by the impact of environmental factors. This is because the RFThermometer needs to get the temperature indicator from RF signals, which largely depend on environmental factors, such as multi-path effect, communication distance, tag position, transmit power. In contrast, Thermotag uses persistence time as the vehicle to sense the temperature, which is related only to the tag's chip circuit and robust to environmental changes.

Table 2: Comparison of Accuracy.

Technique	Mean Error (°C)
DMRT [31]	>10
RFThermometer [33]	5
RTSense [19]	2.9
Thermotag	2.7

So far, we have studied the good performance of Thermotag with a passive tag. Next, we check whether this high performance is scalable and can be generalized to other tags. In Fig. 12, we implement Thermotag on 100 tags and, from those, plot the measurement accuracy of 20 tags. As we can see, the errors are not exactly the same but are all relatively low. About 80% of tags have an error of 3 °C, which means that Thermotag functions well amongst different tags.

The positive experimental results demonstrate that Thermotag can achieve a promising temperature measurement through only a common passive tag, with the competitive advantages of being small-sized, battery-free, and low-cost, which makes item-level temperature sensing possible and practical.

6.3 Impact of Environmental Factors

In this section, we study the impact of several common environmental factors on the performance of Thermotag. We investigate the impact of environmental factors on the temperature estimation results of Thermotag, including the communication distance, tag orientation, communication frequency, transmit power, and target material. All results are evaluated based on the inventoried flag in session 1 (S1) without explicit instructions.

Impact of Communication Distance. The communication distance between a reader and a tag has significant impact on RF signals, e.g., RSSI or the phase value. To investigate the impact of the distance on Thermotag, we vary the communication distance and observe the changes in errors of estimated temperature levels. In this experiment, five distances are tested: 0.2, 0.4, 0.6, 0.8, and 1 m. As shown in Fig. 13, the means of the temperature errors are 1.9 °C, 1.8 °C, 1.5 °C, 2.2 °C, 1.9 °C, respectively, which are small and

bounded within 2.5 °C. These positive results demonstrate that the distance between a reader and a tag has little effect on the performance of Thermotag.

Impact of Tag Orientation. Tag orientation also largely affects the RF-based parameters, e.g., RSSI. In Fig. 14, we study the impact of the tag orientation by observing the persistence time of a tag under different rotation angles, i.e., $0°, 30°, 60°$, $90°$, and $120°$. As we can see, the means of the temperature errors are 2.0 °C, 2.1 °C, 2.9 °C, 2.1 °C, and 2.3 °C, respectively. The consistent results indicate that the estimation error of temperature remains stable, regardless of the tag's rotation angles.

Impact of RF Channels. A typical UHF reader has 16 RF channels working at 920-924 MHz ISM band. RF phase values, a widely used metric for RF sensing, heavily relies on RF channels. To examine whether the channel affects the performance of Thermotag, we validate its measurement performance under five different channels, where $channel_1$ = 920.625 MHz, $channel_2$ = 921.125 MHz, $channel_3$ = 921.625 MHz, $channel_4$ = 922.125 MHz, $channel_5$ = 922.625 MHz. Fig. 15 shows the estimation errors under the five channels. The small difference amongst different channels demonstrates that the proposed system is resistant to the communication channel.

Impact of Transmit Power. Next, we examine the effect of the transmit power of the reader. In this experiment, we set the transmit power to 22, 24, 26, 28, and 30 dBm, respectively, and observe their impact on the estimation accuracy of Thermotag. As shown in Fig. 16, the errors under different transmit powers are small and close to each other. The positive results demonstrate that the transmit power has little impact on our system.

Impact of Target Material. Finally, we study the impact of the material of the tag's surroundings. In the experiment, six liquid materials are tested, including water, beer, Fanta, Coke, sugar, and tea. For each material, we collect 100 samples and calculate the average estimation error of Thermotag. As we can see in Fig. 17, in spite of some differences, the errors of the six liquid materials are all bounded within 4.5 °C. Since the RFID tag is covered by a plastic film, its electronic characters will not largely be affected by the surroundings. The positive results prove that Thermotag can function well for different materials.

6.4 Dense Tag

As shown in Fig. 18, we study whether Thermotag is scalable to the dense-tag case. As aforementioned, given a big tag set, we can partition the set into several small subsets for addressing the congestion problem. In this experiment, we vary the number of tags from 20 to 100 and fix the size of a subset to 10. As can be seen, the number of tags has little impact on the measurement error of temperature. This is because that Thermotag can totally remove the interference of other tags with SL flag when dealing with each subset. The positive results indicate that Thermotag can function well in a dense-tag RFID system.

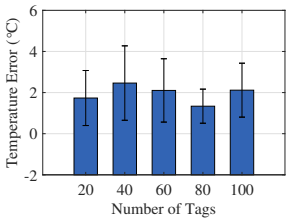

Figure 18: Impact of the number of tags.

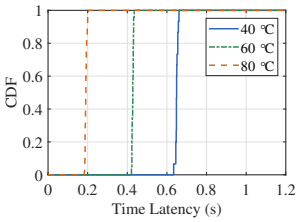

Figure 19: Time efficiency.

6.5 Time Efficiency

High time efficiency is necessary for upper-layer applications. In this experiment, we study the operating time of Thermotag for obtaining a temperature measurement under different temperature values, including 40 °C, 60 °C, and 80 °C. For each temperature, we repeat the temperature measurement 100 times and compute the CDF of the execution time. As shown in Fig. 19, the execution time of Thermotag sees a decline trend as the temperature goes up; this is due to the reduced persistence time that accounts for a large proportion of the execution time. For example, the means of the execution time at 40 °C, 60 °C, and 80 °C are 0.65 s, 0.43 s, and 0.19 s, respectively. This is time-efficient for a majority of RFID-enabled applications.

7 RELATED WORK

Dedicated sensors for temperature detection. Nowadays, there are many varying types of temperature sensors with different features used in daily life [2, 4, 7, 13, 16, 17, 20, 22, 23]. The core attribute of these sensors is that they have some parameters that vary linearly with the temperature; any change of the temperature gives rise to variations in parameters. In general, there are four common types of dedicated sensors: glass thermometer, thermocouple, resistance thermometer, and infrared thermometer. The glass thermometer, the most widely used for temperature measurement, has a glass tube filled with mercury or some other fluid. As temperature rise makes the fluid expand, we can measure the temperature by observing the volume of the fluid [4, 16, 20]. The thermocouple consists of two dissimilar electrical conductors that form an electrical junction. The temperature will develop a temperature-dependent voltage on this junction, which can be used for temperature measurement [13, 23]. Similar to the thermocouple, the resistance thermometer also performs temperature measurement through the temperature related electronic effect. A resistance thermometer is composed of a length of fine wire made of platinum, copper, or any other material that has an accurate resistance-temperature relationship [7, 22]. The resistance of the wire varies as the temperature fluctuates. An infrared thermometer estimates temperature from a portion of the thermal radiation emitted by the object being measured [2, 17, 28]. By checking the amount of infrared energy and emissivity, the object's temperature can be determined. In spite of the wide use of these sensors, they suffer from

three limitations. First, most of them are expensive, making item-level temperature monitoring cost-prohibitive. Second, most existing sensors require external power to activate the internal electronic components, thus leading to a bulky sensor system with high installation or replacement costs. Third, they cannot perform wireless temperature measurement directly; an extra embedded component (e.g., RF transmitter) is required to carry out data transmission [18, 38].

RFID-based temperature sensing. As passive RFID tags have been widely used in a number of applications for good tracking and managing, RFID has become a good choice for item-level temperature sensing due to its compelling features: low cost, small size, and battery-free communication. Early research studies sensor-augmented tags for temperature measurement by combining a common passive tag with a traditional temperature sensor, in which the sensor is in charge of temperature measurement and the tag is responsible for data transmission over the air [5, 6, 10, 21, 24, 25, 27, 30]. One solution is to replace the traditional tag antenna with a specific material that is sensitive to the temperature of its surrounding environment [21, 24]. For example, Shafiq et al. [24] propose a reusable battery-free RFID temperature sensor with a planar dipole antenna that could arise certain frequency shift with temperature. Roth et al. [21] design a temperature activated media for tag antenna that exhibits different colors below and above the selected temperature. However, since no dedicated temperature sensor is used, the resolutions of these sensor tags are usually low.

Another common solution is directly integrating temperature sensors into the RFID chip circuit [10, 27, 30, 37]. Vaz et al. [30] provide a low-power design of a temperature sensor unit that can function properly in a passive tag. Jauregi et al. [10] present a novel wearable RFID temperature sensor tag that collects energy from body heat. Yin et al. [37] propose a temperature sensor that can well balance the trade-off between accuracy and power consumption. Chowdhury et al. [5] design a temperature sensor by observing the discharging time of an on-chip diode. The follow-up work [6] further improves the performance of the sensor by implementing a first-order Δ-Σ loop and the mean error is reduced to 0.1 °C. These sensor-augmented solutions are feasible but suffer from two limitations. First, they largely increase the cost of a tag. Second, since the temperature sensor largely consumes the energy captured by the tag from electromagnetic waves, the communication range between the reader and the tag declines sharply [10].

Some recent work [19, 31, 33] attempts to sense temperature with commodity RFID devices. The basic idea of these methods is that RF signals modulated by a tag vary with the temperature. By observing the fluctuations of received RF phase values, these methods can estimate the temperature level approximately. Specifically, DMRT [31] leverages the differential minimum threshold power as the metric to do the temperature measurement. However, since the resolution of the threshold power of an RFID reader is low, the measurement error of this approach is usually beyond 10 °C. RFThermometer [33] makes a fresh attempt to sense the temperature by using the phase value of the tag. The phase value is more sensitive to the temperature than DMRT and can achieve better performance with a mean error of about 5 °C. However, as the phase value is sensitive to environmental factors, such as the tag location, RFThermometer can only be used in static scenarios where the tag does not move. Among the solutions presented, RTSense [19] is the most advanced work that uses a pair of tags to eliminate the impact of environmental factors. It reduces the mean error of temperature sensing to about 2.9 °C. However, RTSense requires each tag to carry a custom-designed antenna and thereby cannot be implemented by COTS RFID tags directly. In comparison, Thermotag leverages the discharging time of on-chip capacitor as the metric to perform temperature sensing, which is accurate and robust to environmental factors, with no need for any hardware modifications.

8 DISCUSSION AND CONCLUSION

In this paper, we present a novel tag-based thermometer called Thermotag, which uses a common passive RFID tag to sense the temperature for item-level intelligence. The basic idea of Thermotag is that the resistance of a semiconductor diode in a tag's chip is temperature-sensitive. By measuring the discharging period through the reverse-polarized diode, we can estimate the temperature indirectly. We propose a standards-compliant measurement scheme of the discharging period by observing the flag bit of the tag's volatile memory. Furthermore, we build a mapping model between the discharging period and temperature for accurate and reliable temperature sensing. Extensive experiments show that Thermotag has a large sensing span ranging from 0 °C to 85 °C and an average measurement error of 2.7 °C.

Finally, we outline avenues for future research. First, Thermotag has relatively high overhead for per-tag calibration. Designing new approaches (e.g., one-point calibration) to reduce this overhead is an important step to improve the scalability. Second, Thermotag uses only one inventoried flag to sense temperature. A standards-compliant tag has four inventoried flags. It is promising to use these flags jointly for improving accuracy. Third, besides temperature, a future direction is to use the similar approach to measure additional parameters, e.g., soil humidity.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers and shepherd for their insightful comments on this paper. This research is financially supported by the National Natural Science Foundation of China (No. 62072231), the Key Program of the National Natural Science Foundation of China (No. 61932013), the U.S. National Science Foundation (No. CNS-1718708), the Fundamental Research Funds for the Central Universities (No. 14380079), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Jia Liu (jialiu@nju.edu.cn) and Lijun Chen (chenlj@nju.edu.cn) are the corresponding authors.

REFERENCES

[1] 2015. *GS1 EPCglobal. EPC radio-frequency identity protocols generation-2 UHF RFID version 2.0.1.*

[2] Dounia Bitar, Aicha Goubar, and Jean-Claude Desenclos. 2009. International travels and fever screening during epidemics: a literature review on the effectiveness and potential use of non-contact infrared thermometers. *Eurosurveillance* 14, 6 (2009), 19115.

[3] Theodore F Bogart, Jeffrey S Beasley, and Guillermo Rico. 2004. *Electronic devices and circuits*. Pearson/Prentice Hall New Jersey.

[4] Dario Camuffo and Chiara Bertolin. 2012. The earliest spirit-in-glass thermometer and a comparison between the earliest CET and Italian observations. *Weather* 67, 8 (2012), 206–209.

[5] Golam Chowdhury and Arjang Hassibi. 2012. An on-chip temperature sensor with a self-discharging diode in 32-nm SOI CMOS. *IEEE Transactions on Circuits and Systems II: Express Briefs* 59, 9 (2012), 568–572.

[6] Golam Chowdhury and Arjang Hassibi. 2018. An on-chip CMOS temperature sensor using self-discharging PN diode in a Δ-Σ loop. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 6 (2018), 1887–1896.

[7] Jiong Ding, Suijun Yang, and Shuliang Ye. 2018. A fast-multi-channel sub-Millikelvin precision resistance thermometer readout based on the round-robin structure. *Measurement Science Review* 18, 4 (2018), 138–146.

[8] John D Hyde. 2006. Method and apparatus for controlled persistent ID flag for RFID applications. US Patent.

[9] Impinj. [n.d.]. Impinj Monza4 chip datasheet. *https://support.impinj.com* ([n. d.]).

[10] Inigo Jauregi, Hector Solar, Andoni Beriain, Ibon Zalbide, Ainara Jimenez, Inaki Galarraga, and Roc Berenguer. 2016. UHF RFID temperature sensor assisted with body-heat dissipation energy harvesting. *IEEE Sensors Journal* 17, 5 (2016), 1471–1478.

[11] Haojian Jin, Jingxian Wang, Zhijian Yang, Swarun Kumar, and Jason Hong. 2018. Wish: Towards a wireless shape-aware world using passive RFIDs. In *Proc. of ACM MobiSys*. 428–441.

[12] Wenhui Ju. 2016. Study on fire risk and disaster reducing factors of cotton logistics warehouse based on event and fault tree analysis. *Procedia Engineering* 135 (2016), 418–426.

[13] Thomas W Kerlin and Mitchell Johnson. 1999. *Practical thermocouple thermometry*. Instrument Society of America.

[14] Jia Liu, Feng Zhu, Yanyan Wang, Xia Wang, Qingfeng Pan, and Lijun Chen. 2017. RF-Scanner: Shelf scanning with robot-assisted RFID systems. In *Proc. of IEEE INFOCOM*. 1–9.

[15] Zhigang Liu and Andrew K. Kim. 2016. Review of Recent Developments in Fire Detection Technologies. *Journal of Fire Protection Engineering* 13, 2 (2016), 129–151.

[16] Chas G Maier. 2002. Resistance thermometers for chemists. *The Journal of Physical Chemistry* 34, 12 (2002), 2860–2868.

[17] Takashi Matsukawa, Makoto Ozaki, Tomoki Nishiyama, Makoto Imamura, and Teruo Kumazawa. 2000. Comparison of infrared thermometer with thermocouple for monitoring skin temperature. *Critical Care Medicine* 28, 2 (2000), 532–536.

[18] Dae Heon Park, Beom Jin Kang, Kyung Ryong Cho, Changsun Shin, Sungeon Cho, Jang Woo Park, and Won Mo Yang. 2011. A study on greenhouse automatic control system based on wireless sensor network. *Wireless Personal Communications* 56, 1 (2011), 117–130.

[19] Swadhin Pradhan and Lili Qiu. 2020. RTSense: passive RFID based temperature sensing. In *Proc. of ACM SenSys*. 42–55.

[20] Hector Rodriguez, Margaret Williams, John S Wilkes, and Robin D Rogers. 2008. Ionic liquids for liquid-in-glass thermometers. *Green Chemistry* 10, 5 (2008), 501–507.

[21] Mark W Roth. 2019. Dual passive technology RFID temperature activated media. US Patent.

[22] R Saarimaa and P Wallin. 1976. Electronic liquid-in-glass thermometer. *Review of Scientific Instruments* 47, 2 (1976), 195–197.

[23] Utpal Sarma and P Kr Boruah. 2010. Design and development of a high precision thermocouple based smart industrial thermometer with on line linearisation and data logging feature. *Measurement* 43, 10 (2010), 1589–1594.

[24] Yousuf Shafiq, John S Gibson, Hyun Kim, Cedric P Ambulo, Taylor H Ware, and Stavros V Georgakopoulos. 2019. A reusable battery-free RFID temperature sensor. *IEEE Transactions on Antennas and Propagation* 67, 10 (2019), 6612–6626.

[25] Yousuf Shafiq, Julia Henricks, Cedric P Ambulo, Taylor H Ware, and Stavros V Georgakopoulos. 2020. A passive RFID temperature sensing antenna with liquid crystal elastomer switching. *IEEE Access* 8 (2020), 24443–24456.

[26] Longfei Shangguan, Zheng Yang, Alex X Liu, Zimu Zhou, and Yunhao Liu. 2015. Relative Localization of RFID Tags using Spatial-Temporal Phase Profiling. In *Proc. of USENIX NSDI*. 251–263.

[27] Jun Tan, Muralikrishna Sathyamurthy, Alexander Rolapp, Jonathan Gamez, Eckhard Hennig, Eric Schäfer, and Ralf Sommer. 2019. A Fully Passive RFID Temperature Sensor SoC With an Accuracy of $\pm 0.4^{\circ}$C (3σ) From 0°C to 125°C. *IEEE Journal of Radio Frequency Identification* 3, 1 (2019), 35–45.

[28] CG Teran, J Torrez-Llanos, TE Teran-Miranda, C Balderrama, NS Shah, and P Villarroel. 2012. Clinical accuracy of a non-contact infrared skin thermometer in paediatric practice. *Child: Care, Health and Development* 38, 4 (2012), 471–476.

[29] Deepak Vasisht, Guo Zhang, Omid Abari, Hsiao-Ming Lu, Jacob Flanz, and Dina Katabi. 2018. In-body backscatter communication and localization. In *Proc. of ACM SIGCOMM*. 132–146.

[30] Alexander Vaz, Aritz Ubarretxena, Ibon Zalbide, Daniel Pardo, Héctor Solar, Andrés Garcia-Alonso, and Roc Berenguer. 2010. Full passive UHF tag with a temperature sensor suitable for human body temperature monitoring. *IEEE Transactions on Circuits and Systems II: Express Briefs* 57, 2 (2010), 95–99.

[31] Ju Wang, Omid Abari, and Srinivasan Keshav. 2018. Challenge: RFID hacking for fun and profit. In *Proc. of ACM MobiCom*. 461–470.

[32] Ju Wang, Liqiong Chang, Shourya Aggarwal, Omid Abari, and Srinivasan Keshav. 2020. Soil moisture sensing with commodity RFID systems. In *Proc. of ACM MosiSys*. 273–285.

[33] Xiangyu Wang, Jian Zhang, Zhitao Yu, Eric Mao, Senthilkumar CG Periaswamy, and Justin Patton. 2019. RFThermometer: A temperature estimation system with commercial UHF RFID tags. In *Proc. of IEEE ICC*. 1–6.

[34] Erica Weir and Kathy Hatch. 2004. Preventing cold chain failure: vaccine storage and handling. *Cmaj* 171, 9 (2004), 1050–1050.

[35] Binbin Xie, Jie Xiong, Xiaojiang Chen, Eugene Chai, Liyao Li, Zhanyong Tang, and Dingyi Fang. 2019. Tagtag: material sensing with commodity RFID. In *Proc. of ACM SenSys*. 338–350.

[36] Lei Yang, Yekui Chen, Xiangyang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proc. of ACM MobiCom*. 237–248.

[37] Jun Yin, Jun Yi, Man Kay Law, Yunxiao Ling, Man Chiu Lee, Kwok Ping Ng, Bo Gao, Howard C Luong, Amine Bermak, Mansun Chan, et al. 2010. A system-on-chip EPC Gen-2 passive UHF RFID tag with embedded temperature sensor. *IEEE Journal of Solid-State Circuits* 45, 11 (2010), 2404–2420.

[38] Qinghua Zhang, Yi Wang, Guoquan Cheng, Zhuan Wang, and Dongmei Shi. 2014. Research on warehouse environment monitoring system based on wireless sensor network. In *Proc. of IEEE IEA*. 1639–1644.

[39] Yan Zhang, Laurence T Yang, and Jiming Chen. 2009. *RFID and Sensor Networks: Architectures, Protocols, Security, and Integrations*. CRC Press.