

# A Framework for Distributed Managing Uncertain Data in RFID Traceability Networks

Jiangang Ma<sup>1</sup>, Quan Z. Sheng<sup>1</sup>, Damith Ranasinghe<sup>1</sup>,  
Jen Min Chuah<sup>1</sup>, and Yanbo Wu<sup>2</sup>

<sup>1</sup> School of Computer Science, The University of Adelaide, Australia  
jiangang.ma@adelaide.edu.au

<sup>2</sup> Beijing Jiaotong Univeresity, Beijing, 100044, China  
ybwu@bjtu.edu.cn

**Abstract.** The ability to track and trace individual items, especially through large-scale and distributed networks, is the key to realizing many important business applications such as supply chain management, asset tracking, and counterfeit detection. Networked RFID (radio frequency identification), which uses the Internet to connect otherwise isolated RFID systems and software, is an emerging technology to support traceability applications. Despite its promising benefits, there remains many challenges to be overcome before these benefits can be realized. One significant challenge centers around dealing with *uncertainty* of raw RFID data. In this paper, we propose a novel framework to effectively manage the uncertainty of RFID data in large scale traceability networks. The framework consists of a global object tracking model and a local RFID data cleaning model. In particular, we propose a Markov-based model for tracking objects globally and a particle filter based approach for processing noisy, low-level RFID data locally. Our implementation validates the proposed approach and the experimental results show its effectiveness.

**Keywords:** RFID, Internet of Things, Uncertainty, Traceability Networks.

## 1 Introduction

Radio Frequency Identification (RFID) is a wireless communication technology that is useful for identifying objects. RFID uses radio-frequency waves to transfer identifying information between tagged objects and readers without line of sight, thus enabling automatic identification [1]. The ability to track and trace individual items—especially through large-scale and distributed networks—is the key to realizing many important business applications such as supply chain management, asset tracking, and counterfeit detection [2–5].

One of the important technological advances that targets large-scale traceability (e.g., nation-wide supply chain management across companies) is the so-called “Networked RFID” [6]. The basic idea behind Networked RFID is to use the Internet to connect otherwise isolated RFID systems and software. With Networked RFID, traceability applications analyze automatically recorded identification events to discover the current location of an individual item. They can also retrieve the historical information, such as previous locations, time of travel between locations, and time spent in storage.

Such technological advances will revolutionize our ability to monitor the world around us, allowing critical decisions and required interventions to be made in a timely fashion.

While RFID provides promising benefits in many applications, there remains significant challenges to be overcome before these benefits can be realized. Central to these challenges is the *uncertainty* of the data collected by the underlying RFID networks. Due to the sensitivity of sensing to the orientation of reading, interference, malfunction of reading components, and many environmental factors, RFID data are typically incomplete, imprecise, and even misleading [7, 8]. Obviously, when such data streams are used directly in monitoring and tracking applications (e.g., product recall), the quality of the applications can be a significant concern.

The inherent uncertainty of the raw RFID data makes it impossible to be used directly in high-level applications. Instead, sophisticated approach needs to be developed to be able to support uncertainty as a first-class citizen in RFID applications. In this paper, we design and implement a novel framework for uncertainty management in large scale RFID traceability networks, geared towards efficiently and accurately supporting traceability applications. In particular, we develop techniques that cope with ambiguous and imprecise RFID data by transforming low-level RFID readings into probabilistic events. In a nutshell, our key contributions are as the following:

- We introduce a data model for globally tracking moving objects. The proposed approach is based on a Markov-based process to infer objects' actual location according to probabilistic RFID observations.
- We propose a sampling-based inference technique to capture the uncertainty of RFID raw data. The technique also produces the probability distribution of RFID objects from dynamic and noisy low-level RFID data.
- We validate the proposed techniques in prototype implementation and the experimental results show the effectiveness of the proposed techniques.

The remainder of this paper is organized as follows. In Section 2, we discuss some background information related to managing uncertain data in RFID traceability networks. In Section 3, we introduce a data model for tracking moving objects and in Section 4, we discuss local RFID data management. In Section 5, we present our experiments to show the effectiveness and efficiency of our approach. Section 6 is dedicated to the related work and Section 7 concludes the paper and discusses some future research directions.

## 2 Preliminaries

In this section, we first describe a scenario of recalling problematic drugs in a supply chain application. We then propose an overall architecture and show how to efficiently manage, query and analyze uncertain information obtained from RFID data sources.

### 2.1 An Example of Uncertainty in Object Tracking

We first use a simple example to illustrate locating and recalling problematic products in a supply chain network, as shown in Figure 1. Suppose a particular type of drug to be distributed from its manufacturer `Brisbane Plant` in Brisbane to its destination

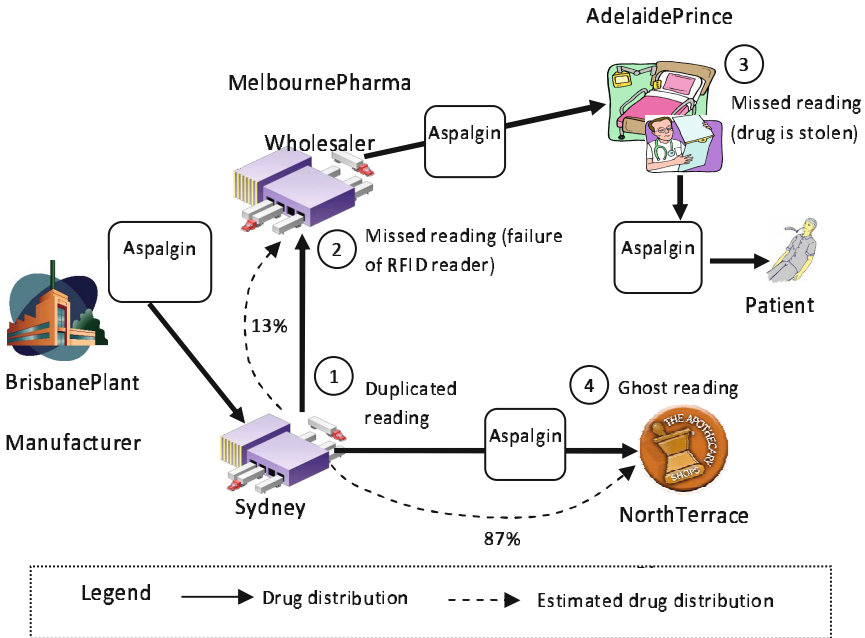


Fig. 1. An example of uncertainty in an RFID-enabled supply chain

Royal Adelaide Hospital in Adelaide. On the path of distribution of the drug, RFID readers are positioned at different nodes like Melbourne Wholesaler and Sydney Distributor to detect the event when the drugs pass by. Further suppose that problematic drugs are detected at the Royal Adelaide Hospital in Adelaide. In this case, the source in Brisbane of problematic drugs needs to be identified. Having tracked the source, the system may need to find similar problematic drugs that may be distributed in other places (e.g., warehouses and pharmacies).

Unfortunately, due to the sensitivity of sensing to the orientation of readings, interference, malfunction of reading components, and many environmental factors, raw RFID data are highly noisy [7, 8]. For example, there could be duplicated reads, missed readings (either due to malfunction of an RFID reader, or due to the objects being stolen or misplaced), and even “ghost” reads, meaning that a tag “captured” by an RFID reader does not exist or is not within the reader’s detecting field. Obviously, when such data streams are used directly in monitoring and tracking applications (e.g., product recall), the quality of the applications can be a significant concern.

## 2.2 Overview of the Framework

In this section, we provide a high level overview of the proposed traceability framework. The framework involves two main tracking components: a global object tracking component and a local RFID data management component.

- The global object tracking component (GOTC), as shown in Figure 2, is used to support tracking and tracing the objects moving from one organization to the other. Our traceable network includes three main entities: sensor nodes, receptors and objects. First, a node is equipped with battery-powered devices that have basic sensing modules and communication interfaces. In particular, each node governs a number of receptors (e.g., RFID readers) whose signals can cover and monitor some areas. These nodes are deployed and dispersed in different sites for monitoring and tracking objects. For example, in a supply chain network, a node may be a distribution center, or a retail store and RFID readers are deployed at the fixed locations such as the entrances of a warehouse. Second, an object is any monitored entry attached with RFID tags such as items and goods, etc. Very often, objects move across the nodes confined within this traceability network. We will use a Markov-based process to model and infer objects actual locations based on probabilistic observations, which will be detailed in the next section.
- Local RFID Data Management (LRDM), as shown in Figure 3, represents the internal structure and functionalities of each node in a traceability network. LRDM component is executed at individual sensor node. To capture uncertainty of raw RFID data, we employ a sampling-based inference technique called particle filtering [9] to compute the probability distribution of objects from dynamic and noisy low-level RFID data. The generated probabilistic RFID data is then processed in order to extract the high-level events that can be used by applications. In addition, the processed data is stored in databases in terms of records according to a data model developed in the framework, readily for querying and mining.

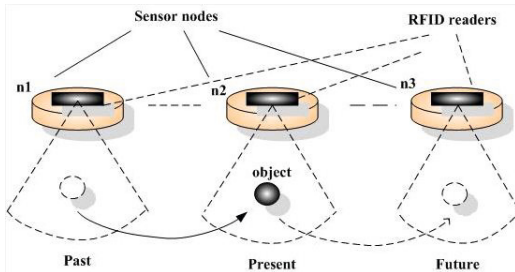


Fig. 2. Global object tracking

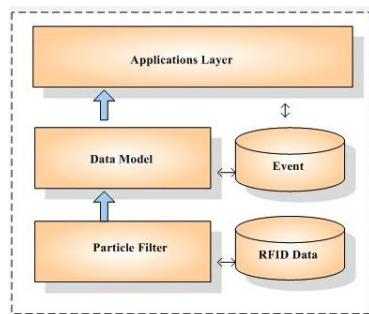


Fig. 3. Local data management

### 3 A Global Data Model for Tracking Moving Objects

In this section, we introduce a data model for globally tracking moving objects. The proposed approach is based on the Markov-based process to infer objects' actual location according to probabilistic RFID data observations.

### 3.1 Location Information Flows vs. Observation Information Flows

We start with the basic description of tracking approach in traceable networks, shown in Figure 2. When tracking and tracing objects that move along a certain route, we are interested in two types of information flows: *the location information flows* (LIF) and *the observation information flows* (OIF). Physically, one object moves along a certain route (e.g., a pallet moves in a supply chain network) and such movement of objects causes the change of object's physical locations. We call this LIF. On the other hand, once the objects are captured by RFID readers, the movement of objects generates digitalized information, we call this OIF.

However, due to different factors such as inaccurate objects identifications and missed readings caused by RFID readers, the observed information (OIF) may not exactly reflect the actual location of the object (LIF), which generates uncertain information. In addition, LIF are often unobservable state variables due to the uncertainty, so we refer LIF as hidden state variables. On the other hand, OIF are observable evidence variables (e.g., observed by RFID readers) [10]. Thus, in this paper, one of our aims is to model and reason objects' actual locations according to OIF. We deal with this problem using a Markov-based probability approach.

### 3.2 A Markov-Based Model for Global Object Tracking

Formally, we consider a distributed traceability network, comprising a collection of  $n$  nodes. The LIF for the nodes are represented as  $\mathcal{X}=\{x_1, x_2, \dots, x_n\}$ , which are often unobservable due to the uncertainty.  $\mathcal{Y}=\{y_1, y_2, \dots, y_n\}$  denotes a set of OIFs that are generated by RFID readers. We use  $\mathcal{X}_t$  to denote unobservable state variable at time  $t$  and  $\mathcal{Y}_t$  for observable evidence variable at time  $t$ .

We now consider an object moving across a traceability network as a first-order Markov random process. At a given point of time, although an object can be at any location of a set of possible locations  $\mathcal{X}$ , the first-order Markov process assumes that the probability distribution at  $t + 1$  is entirely determined by the current state at  $t$ . Thus, we use a conditional probability  $p(\mathcal{X}_t|\mathcal{X}_{t-1})$  to represent the location change of an object from state  $\mathcal{X}_{t-1}$  to  $\mathcal{X}_t$ . The transition probability  $p(\mathcal{X}_t|\mathcal{X}_{t-1})$  can be further represented by a matrix  $\mathcal{M}$  as the following:

$$M_L = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & & \vdots \\ m_{n1} & \cdots & m_{nn} \end{bmatrix} = \begin{bmatrix} p(x_1|x_1) & \cdots & p(x_n|x_1) \\ \vdots & p(x_j|x_i) & \vdots \\ p(x_1|x_n) & \cdots & p(x_n|x_n) \end{bmatrix}$$

where the entry  $m_{ij} = p(x_j|x_i)$  represents the transition probability by which state  $x_i$  changes to state  $x_j$ . Based on the above transition model, an object should be in a certain location after state  $x_i$ , so  $\sum_{j \neq i} m_{ij} = \sum_{j \neq i} p(x_j|x_i) = 1$  should hold.

Similarly, we can further model the relationships between location flows  $\mathcal{X}$  and observation flows  $\mathcal{Y}$ . Intuitively, the observable variables  $\mathcal{Y}$  are often affected by actual state variables  $\mathcal{X}$ . In other words, state variables  $\mathcal{X}$  cause the observable variables  $\mathcal{Y}$  to take different values. Formally, let  $\mathcal{Y}$  be a set of the observable variables taking values from RFID readers. Based on Markov assumption, we can use a conditional distribu-

tion to describe how state variables  $\mathcal{X}$  affect the observable variables  $\mathcal{Y}$ . This model is represented by a matrix  $\mathcal{N}$ :

$$N = \begin{bmatrix} n_{11} & \cdots & n_{1m} \\ \vdots & & \vdots \\ n_{n1} & \cdots & n_{nm} \end{bmatrix} = \begin{bmatrix} p(y_1|x_1) & \cdots & p(y_m|x_1) \\ \vdots & p(y_j|x_i) & \vdots \\ p(y_1|x_m) & \cdots & p(y_m|x_m) \end{bmatrix}$$

where the entry  $n_{ij} = p(y_j|x_i)$  represents the probability that system will demonstrate observation value  $y_j$  given that the object is in location  $x_i$ . Based on the above transition model, an object should be in some location after state  $x_i$ , so  $\sum_{j \neq i} m_{ij} = \sum_{j \neq i} p(x_j|x_i) = 1$  should hold. An example of transition and observation model for an RFID traceability network is shown in Figure 4.

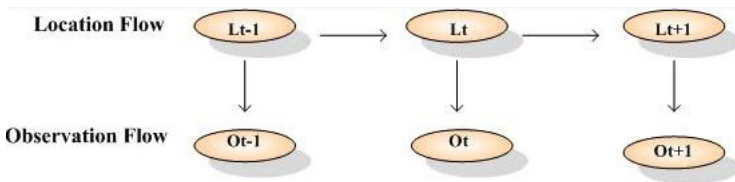


Fig. 4. Transition model for RFID traceable network

## 4 Local RFID Data Management in Traceable Networks

In this section, we describe our local RFID data management in traceable networks. We first describe a probabilistic model of uncertain data. We then present a sampling-based inference technique to capture uncertainty of RFID raw data.

### 4.1 A Probabilistic Model for Capturing Uncertain Data

In RFID-based tracking applications, raw data often exhibit uncertainty and impreciseness. For example, the products may be stolen, counterfeited, damaged, and misplaced. Thus, data produced by sensors (e.g., RFID readers) are often noisy. These uncertainties can be further classified into four categories: *false negative*, *true negative*, *false positive*, and *duplicate readings*. To capture such uncertainty in data, we use a continuous random variable  $x$  and describe data’s uncertainty by using probability density functions  $pdf(x)$ .

**Definition 1.** (*Uncertain Object*) An uncertain object  $o$  is defined to be a point  $x$  within a region  $\mathcal{R}_u$  covered by RFID readers. The positions of the object  $o$  follow a probability density function  $pdf(x)$ , denoting the likelihood of the object’s position, and the condition holds:  $\int_{x \in \mathcal{R}_u} pdf(x)dx = 1$ .

We further use Bayes rule to infer an object’s location. As mentioned above, objects’ locations in RFID applications are regarded as hidden variables  $x$  and observation variables  $y$  are the readings produced by RFID readers. Thus, our aim is to compute the joint

probability distribution  $p(x, y)$  over both hidden and observed variables. In addition, given an observed values  $y$ , this joint model induces the computation of a conditional distribution  $p(x|y)$ , which can be used to predict the objects' locations. More precisely, based on Bayes' rule, the conditional distribution  $p(x|y)$  can be represented as:

$$f(x|y) = \frac{f(y|x)f(x)}{\int f(y|x)f(x)dx}$$

Suppose we have  $n$  observations  $\mathcal{Y}^n = (\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n)$  of RFID data at each time-step, we replace  $f(y|x)$  with:

$$f(y_1, y_2, \dots, y_n|x) = \prod_{i=1}^n f(y_i) = \Gamma(x)$$

Thus, the current true state is defined by the posterior density over the random variable  $x$  conditioned on all RFID data:

$$f(x|y^n) = \frac{f(y^n|x)f(x)}{\int f(y^n)f(x)dx} = \frac{\Gamma(x)f(x)}{h_n}$$

where

$$h_n = \int \Gamma(x)f(x)dx$$

is the normalizing constant. In this case, one location variable in  $\mathcal{X} = (x_1, \dots, x_k)$  (e.g.,  $x_1$ ) is determined by computing the following marginal posterior density:

$$f(x_1|\mathcal{Y}^n) = \int \int \dots \int f(x_1, \dots, x_k|\mathcal{Y}^n)dx_2 \dots dx_k$$

The challenge here is to perform accurate inference for a large number of RFID data because the cost of computing such posterior densities grows exponentially over time. In order to deal with this issue, we next introduce a sampling-based inference called particle filtering to approximate the target conditional distribution.

## 4.2 Processing Uncertain RFID Data Based on Particle Filtering

The main idea of the particle filter [9, 11] is to approximately represent the posterior probability by a set of limited random state samples (also called particles), extracted from this posterior probability. One of the main advantages for particle filtering is that such approximation of original conditional probability is able to achieve the expected solutions with only a limited number of samples. It can also represent a broader range of distributions than common probabilistic distributions like Gaussian Distribution.

Intuitively, the main task of a tracking application in RFID is to estimate states (e.g., the location of an object), based on the available evidence states (e.g., RFID readings). The estimating inference includes two main steps: *prediction* and *importance sampling*. The prediction stage is to construct the set of candidate particles, while the update stage is to construct the set of qualified particles from candidate particles. These qualified particles will approximate the original posterior *pdf*.

**Algorithm 1.** State Update

---

**Input:** raw RFID readings,  $\mathbf{X} = \{x^i | i = 1 \dots I\}$  **Output:** estimation of an object's location

---

1. **for**  $\forall x^i \in \mathbf{X}$  **do**
  2.    $x^i\_location + = motion\_model.$
  3.    $x^i\_location + = diffuse\_rate * random\_number.$
  4. **end for**
  5.  $prob \leftarrow measure\_probability(x^i\_location).$
  6. calculate normalization\_value
  7. **for**  $\forall x^i \in \mathbf{X}$  **do**
  8.    $x^i\_prob / = normalization\_value.$
  9.    $calculate\ x^i\_prob.cumulative\_prob$
  10. **end for**
  11. **for**  $\forall x^i \in \mathbf{X}$  **do**
  12.    $r \leftarrow random\_num\_coeffie.$
  13.    $p \leftarrow first\ x^i\_prob.cumulative\_prob \geq r$
  14.    $ressampled\_x^i \leftarrow p$
  15. **end for**
- 

**Algorithm 2.** Sampling Process

- 
1. **if** object reading is missing
  2.   **for**  $\forall x^i \in \mathbf{X}$  **do**
  3.      $find\_nearest\_reader(x^i).$
  4.      $x^i \leftarrow position\_of\_nearest\_sensor$
  5.      $d \leftarrow x^i\_distance\_from\_reader$
  6.   **end for**
  7. **else**
  8.   **for**  $\forall x^i \in \mathbf{X}$  **do**
  9.      $dis \leftarrow compute\_distance(x^i - reader).$
  10.   **end for**
  11. **end if**
- 

The input to the particle filter algorithm are a set of samples (states) of an object,  $\mathcal{X}_{t-1} = \{x_{t-1}^1, x_{t-1}^2, \dots, x_{t-1}^m\}$ , and a set of current observation  $y_t$ . At the first step of constructing set  $\mathcal{X}_t^c$  of candidate particles, the algorithm processes each particle in  $\mathcal{X}_{t-1}$ . At the end of this step, the set  $\mathcal{X}_t^c$  of candidate particles includes a set of weighted particles  $\mathcal{X}_t^c = \langle x_t^i, w_t^i \rangle$ , where  $w_t^i$  is the weight of a particle  $x_t^i$  and  $0 \leq i \leq m$ . At the second step, the algorithm selects the particles from  $\mathcal{X}_t^c$ , and puts the particles that have higher weights as qualified particles into the set  $\mathcal{X}_t^q$  of qualified particles. At the end of this step, the set  $\mathcal{X}_t^q$  includes all qualified particles. Finally, these qualified particles in the set  $\mathcal{X}_t^q$  approximate the original posterior probabilistic distribution *pdf*. The particle filtering algorithm is a recursive process that includes constructing set  $\mathcal{X}_t^c$  of candidate particles, and the set  $\mathcal{X}_t^q$  of qualified particles.



*Prediction.* This step updates the particles to reflect the new states of the observed objects (see Algorithm 1). A new state  $x_t$  at time  $t$  will be generated based on the particle  $x_{t-1}$  and the observation  $y_t$ . In our implementation, we define the difference between the location of readers and object's previous location as the motion model. The motion model describes how the states of an object change. Thus, in the prediction stage, we can update each of the particles based on the results from the motion model.

---

**Algorithm 3.** Managing Missing Readings
 

---

```

1. read input  $\mathbf{X} = \{x^i | i = 1 \dots I\}$ 
2. do
3.   get readings
4.   if reading_time = current_time
5.     if object  $\neq$  exist
6.       initialize_object.
7.     else
8.       get_object
9.     end if
10.    calculate_motion_model
11.    update_state_of_object
12.  else
13.     $\mathbf{M} = \{m^i | i = 1 \dots H\} \setminus *check\_missing\_readings$ 
14.    for  $\forall m^i \in \mathbf{M}$  do
15.      simulate_uncertainty.
16.      update_object_state.
17.    end for
18.  end if
19. while ! more readings
  
```

---

*Sampling.* This step involves taking samples for transition distribution  $p(x_t | y_t, x_{t-1})$ . In our case, the probability of each particle is calculated using a standard normal distribution (Gaussian)  $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ , where  $\mu$  is mean, and  $\sigma^2$  variance. Then for each particle  $x_t^i$ , we compute its weight  $w_t^i = p(y_t | x_t^i)$ . For example, if an observed object is within the range that the RFID reader covers, the particles produced by the object are assigned higher weights because these particles have a higher likelihood of representing the correctly estimated locations. At the end of this step, the set  $\mathcal{X}_t^c$  of candidate particles includes a set of weighted particles  $\mathcal{X}_t^c = \langle x_t^i, w_t^i \rangle$ . An algorithm of sampling process is shown in Algorithm 2.

*Resampling.* This step converts the set  $\mathcal{X}_t^c$  of candidate particles into the set  $\mathcal{X}_t^q$  of qualified particles. However, only those particles in set  $\mathcal{X}_t^c$  that have bigger values of weights are selected and transformed into  $\mathcal{X}_t^q$ .

We also design an algorithm (see Algorithm 3) to deal with missed readings in order to further improve the effectiveness of our approach. When the readings are missed, the location of an object indicates uncertainty. To simulate the uncertainty of an object's

locations due to missed readings, we set the locations of some particles randomly, and estimate the locations of the remaining particles according to  $p(X_t|X_{t-1})$ .

### 4.3 A Data Store Model

We develop a new schema to store cleaned RFID data for supporting querying and mining. Our model incorporates RFID data’s uncertain information, including time, tagID, location, and probability. RFID records form a probabilistic database **PDB**:

**Definition 2.** Each record  $r \in \mathbf{PDB}$  is represented by a tuple  $r = \langle t^r, d^r, l^r, p^r \rangle$ , where  $t^r, d^r, l^r, p^r$  denotes the ‘time’, ‘tagID’, ‘location’, ‘probability’.

For instance, one instance of record (2:05pm, EPC0001, Sydney-D, 0.57) indicates that at 2:05pm, the object with ID of EPC0001 was located at Sydney Distributor (i.e., Sydney-D) with the probability of 0.57. This model also stores the probability distribution over an RFID object’s location at a given time. For example, there might be another tuple (2:05pm, EPC0001, Melbourne-W, 0.43), indicating that there was also a 0.43 probability that the object with ID of EPC0001 was at Melbourne Wholesaler (i.e., Melbourne-W). An example of RFID data records is shown in Table 1.

Table 1. RFID data records

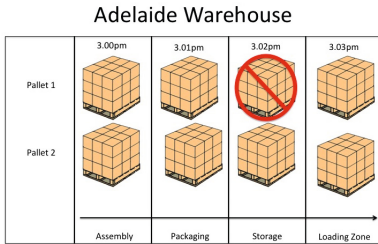
Raw Data Records	$(t_1, r_1, l_1), (t_1, r_2, l_1), (t_1, r_3, l_2), (t_2, r_2, l_1)$ $(t_2, r_1, l_3), (t_3, r_1, l_2), (t_4, r_4, l_1), (t_4, r_3, l_3)$ $(t_2, r_3, l_4), (t_5, r_3, l_4), (t_5, r_1, l_3), (t_5, r_4, l_5)$
Probabilistic RFID Data Records	$(t_1, r_2, l_1, 0.1), (t_1, r_3, l_2, 0.4), (t_2, r_2, l_1, 0.5)$ $(t_2, r_1, l_3, 0.4), (t_3, r_1, l_2, 0.2), (t_4, r_4, l_1, 0.3)$ $(t_2, r_3, l_4, 0.3), (t_5, r_3, l_4, 0.4), (t_5, r_1, l_3, 0.6)$

## 5 Experiments

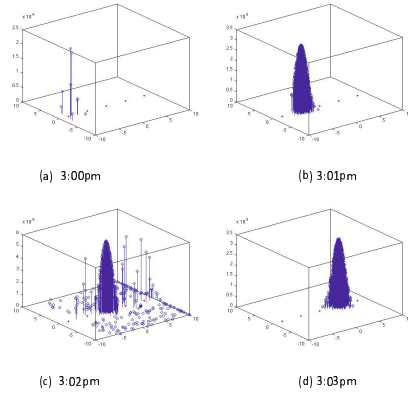
To validate the proposed approach, we conducted two groups of experimental studies. The first group tested the effectiveness of the basic functionality of the system, and the second group explored the accuracy of the particle filtering-based approach.

We use NetBeans to develop two Java applications to support experiments. The first Java application is an RFID data generator that produces a series of simulated RFID readings. The generated RFID data are then used as the input of the particle filtering system. The probabilistic outputs generated from the particle filtering system are fed into the second Java application. This Java application translates probabilistic outputs into a SQL file for supporting updating and querying the database.

To test the basic functionality of the system, we have designed a scenario where two pallets travel through different stages within a warehouse. The missed readings for the first pallet occur at 3.02pm shown in Figure 5. The testing results are represented by four three-dimensional graphs, as shown in Figure 6, each representing the first pallet’s predicted location at a given time.



**Fig. 5.** Missed reading in warehouse scenario



**Fig. 6.** The result of testing particle filtering system

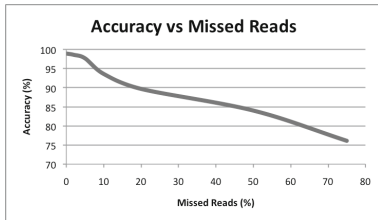
In these graphs, the X axis and Z axis of the visualization represent the X and Y coordinates of the warehouse, while the Y axis represents the probability values. The red dots signify the different RFID readers and their positions within the warehouse. In addition, each blue circle represents a particle of the object observed and the probability of that particle. This experimental results show that the proposed system is able to predict the locations of moving objects. For example, Figure 6 shows that the particles are correctly predicted to be around the third reader with a probability of 0.489, although some readings are missed at 3.02pm.

### 5.1 Experimental Results

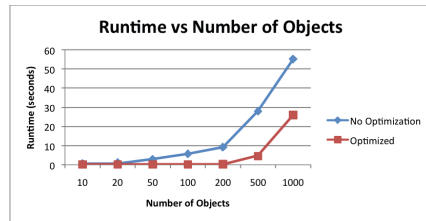
To show the accuracy of our tracking system, we carried out two experimental studies. The first experiment explores the accuracy of particle filtering system whilst the second group demonstrates the effect of optimization on the runtime.

*Accuracy of Particle Filtering System.* In this experiment, we studied the effect of missed readings on the accuracy of location estimation. For each category of missed readings, our data generator generated an input file containing data with a certain percentage of missed readings. Each input data file includes 100 records, and these files were used to examine the average accuracy. Figure 7 shows the experimental results. Overall, our system is able to perform object location estimation with an accuracy of 76%. It indicates a good property of our system even if some readings are missing.

*Effect of Optimization on Runtime.* We have also examined the effect of optimization on the runtime speed. For each experiment with a set of objects, we executed each version of the program five times to determine the average runtime. Our experiments show that our approach can reduce the numbers of objects to be processed. This also has an impact on the runtime of the particle filter algorithm. From Figure 8, we can see that when the number of objects exceed 1,000, the time used in our approach is much



**Fig. 7.** Accuracy vs percentage of missed reads



**Fig. 8.** Runtime speedup

less than the time used without our approach. Therefore, together with the ability to perform object location estimation in reduced time, we can conclude that our system is suitable for large-scale practical applications.

## 6 Related Work

Managing uncertain data is an important research topic in RFID-based applications. There are a number of researches focusing on RFID networks [12–14], uncertain data management [7, 15, 16], data model [17, 18] and probabilistic databases [4, 19–21]. However, there still lacks of effective framework to support uncertain data management in a distributed traceability network.

EPCglobal [7] provides rich standards to support RFID-based tracking networks. Currently, EPCglobal Architecture Framework (EAF) is widely regarded as one of the most well-known RFID network architectures in industry [22]. In addition, works in [13, 23] propose generic approaches for tracing and tracking of objects in large-scale and distributed environments such as the Internet of Things. In particular, by analyzing a wide range of traceability applications, these works provide models for moving objects in discrete spaces, which are mainly built on top of the DHT (Distributed Hash Table) based overlay network. Furthermore, the proposed frameworks separate functionalities into three isolated modules: identity, capture, and exchange.

The work related to uncertain data management can be found in [7, 15, 16]. The work by Diao et al. [16] proposes a probabilistic model and approach to transform raw data streams into tuple streams with quantified uncertainty. This approach approximates object location estimation based on data streams processing, making their research relevant to our approach. However, their techniques cannot be directly used in our work because their work focuses on general estimations for objects tracking and not deal with the location estimation with the large-volume of data in distributed applications such as supply chain networks. To understand the containment relationships of objects for location estimating, the work in [15] presents a SPIRE framework, which relies on packaging level information extracted from tagIDs of objects to form time-varying graphs and depict the relationships of inter-objects' containments. Based on the graphs, estimating locations of an object can be done by inferring the edges and the nodes in the graph, so that a probabilistic distribution over all possible states for each node could be built.

In the area of particle filtering in RFID-based applications, the Cascadia project [17] applies a particle filter approach to infer the probability distributions over an object's location. In particular, each particle is associated with a location. Upon receiving a raw RFID event, the particles are weighted based on the consistency of their coordinates with the raw RFID event. However, since this project targets applications from a single administrative domain, it is not required to process data in a large-scale and widely distributed environments.

## 7 Conclusion

Effectively managing uncertain data in RFID traceability networks still remains a challenge. In this paper, we have studied the main problems related to tracking the movements of objects and particularly focused on processing ambiguous and imprecise RFID data. We have designed and implemented a novel framework, which improves the existing techniques for tracking the movement of objects and cleaning RFID raw data. Within the framework, we use a Markov-based process to infer objects' actual location according to probabilistic RFID data observations. In addition, we propose a sampling-based inference technique called particle filtering to capture uncertainty of RFID raw data, and to compute the probability distribution of RFID objects from dynamic and noisy low-level RFID data. Our probabilistic reasoning approach enables efficient and accurate support for traceability applications. The experimental results have shown that our approach is capable in object estimation with high accuracy and scalable in large-scale applications.

Our further work include conducting more experiments to further study the system performance. We also plan to develop a model for probabilistic RFID event generation. This will particularly consider the issue of Unordered Event Stream (UnES) where the order of arriving RFID events might not match with the order of the occurrence of the events in the real world (e.g., due to network routing delay).

## References

1. Nath, B., Reynolds, F., Want, R.: Rfid technology and applications. *IEEE Pervasive Computing* 5, 22–24 (2006)
2. Wu, Y., Ranasinghe, D.C., Sheng, Q.Z., Zeadally, S., Yu, J.: RFID Enabled Traceability Networks: A Survey. *Distributed and Parallel Databases* 29, 397–443 (2011)
3. Franklin, M., Jeffery, S., Krishnamurthy, S., Reiss, F., Rizvi, S., Wu, E., Cooper, O., Edakkunni, A., Hong, W.: Design considerations for high fan-in systems: The hifi approach. In: *Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research, CIDR 2005* (2005)
4. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing and analyzing massive rfid data sets. In: *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006*, pp. 83–83. *IEEE* (2006)
5. Ilic, A., Andersen, T., Michahelles, F.: Increasing supply-chain visibility with rule-based rfid data analysis. *IEEE Internet Computing* 13, 31–38 (2009)
6. Roussos, G., Duri, S., Thompson, C.: Rfid meets the internet. *IEEE Internet Computing* 13, 11–13 (2009)

7. Jeffery, S., Franklin, M., Garofalakis, M.: An adaptive rfid middleware for supporting meta-physical data independence. *The VLDB Journal* 17, 265–289 (2008)
8. Sheng, Q., Li, X., Zeadally, S.: Enabling next-generation rfid applications: Solutions and challenges. *Computer* 41, 21–28 (2008)
9. Fox, V., Hightower, J., Liao, L., Schulz, D., Borriello, G.: Bayesian filtering for location estimation. *IEEE Pervasive Computing* 2(3), 24–33 (2003)
10. Russell, S.: *Artificial intelligence: A modern approach*, December 30 (2002)
11. Ng, B., Peshkin, L., Pfeffer, A.: Factored particles for scalable monitoring. In: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 370–377 (2002)
12. EPCGLOBAL, <http://www.epcglobal.com>
13. Wu, Y., Sheng, Q., Ranasinghe, D.: P2p object tracking in the internet of things. In: *Proceedings of International Conference on Parallel Processing (ICPP 2011)*, pp. 502–511. IEEE (2011)
14. Cambridge University: Serial-level inventory tracking model. Bridge WP03, Cambridge University, BT Research (2007)
15. Nie, Y., Cocci, R., Cao, Z., Diao, Y., Shenoy, P.: Spire: Efficient data inference and compression over rfid streams. *IEEE Transactions on Knowledge and Data Engineering*, 141–155 (2012)
16. Diao, Y., Li, B., Liu, A., Peng, L., Sutton, C., Tran, T., Zink, M.: Capturing data uncertainty in high-volume stream processing. In: *Proceedings of the 4th Biennial Conference on Innovative Data Systems Research, CIDR 2009* (2009)
17. Welbourne, E., Khossainova, N., Letchner, J., Li, Y., Balazinska, M., Borriello, G., Suciu, D.: Cascadia: A system for specifying, detecting, and managing rfid events. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys 2008)*, New York, USA, pp. 281–294 (2008)
18. Wang, F., Liu, P.: Temporal management of rfid data. In: *Proceedings of International Conference on Very Large Databases (VLDB 2005)*, Norway, pp. 1128–1139 (2005)
19. Soliman, M., Ilyas, I., Chen-Chuan Chang, K.: Top-k query processing in uncertain databases. In: *Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007)*, pp. 896–905. IEEE (2007)
20. Zhang, Y., Lin, X., Zhu, G., Zhang, W., Lin, Q.: Efficient rank based knn query processing over uncertain data. In: *Proceedings of the 26th International Conference on Data Engineering (ICDE 2010)*, pp. 28–39. IEEE (2010)
21. Cormode, G., Garofalakis, M.: Sketching probabilistic data streams. In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 281–292. ACM (2007)
22. EPCglobal: EPCglobal Specifications, <http://www.epcglobalinc.org/standards/specs>
23. Mo, J., Sheng, Q., Li, X., Zeadally, S.: Rfid infrastructure design: a case study of two australian rfid projects. *IEEE Internet Computing* 13, 14–21 (2009)