2010 24th IEEE International Conference on Advanced Information Networking and Applications

# Enabling Scalable RFID Traceability Networks

Quan Z. Sheng, Yanbo Wu, Damith Ranasinghe

School of Computer Science
The University of Adelaide
Adelaide, SA 5005, Australia
Email: {qsheng, yanbo, damith}@cs.adelaide.edu.au

## Abstract

*The ability to track individual objects is essential to many aspects of our modern life such as product recalls and anti-counterfeiting. As a non-line-of-sight technology, radio-frequency identification (RFID) provides an effective way to record movements of objects and has recently emerged as an enabling technology for traceability applications. Unfortunately, realizing RFID traceability networks in large-scale, distributed environments brings many fundamental research and development issues. In particular, applications will generate an unprecedented amount of transactions and data that requires novel approaches not only in RFID data processing and management, but infrastructure and architecture design. In this paper, we describe our approach for realizing a scalable RFID traceability network, which can efficiently and effectively support traceability applications. With a novel data model, traceability applications can share data across independent organizations in a peer-to-peer fashion.*

## I. Introduction

Traceability refers to the capability of an application to track the state (e.g. location, temperature) of goods, discover information regarding its past state and potentially estimate its future state. Traceability is essential to a wide range of applications such as manufacturing control, logistics of distribution, product recalls, and anti-counterfeiting [1], [2], [3]. For example, the Australian Government Department of Health and Aging recently reported that annually 5.4 million cases of food-borne illness occur in Australia, which leads to 2.1 million days of lost work, 1.2 million people to visit a doctor, and 120 deaths every year[1]. These have revealed the urgent need for improved ways of locating and recalling problematic products (e.g., contaminated food or counterfeited drugs) that have been released into the community.

Radio-frequency identification (RFID) is a wireless communication technology that is useful for precisely identifying objects. RFID uses radio-frequency waves to transfer identifying information between tagged objects and readers without line of sight, making automatic tracking and tracing possible [2], [3]. For example, in supply chain management, RFID tags are used to track products from supplier to distribution center, warehouse stock, and point of sale. In a typical future deployment, RFID tags will be affixed to individual product items, and companies install RFID readers at various locations on their premises to capture tag reading events. As the tagged items are transported across companies, trails of tag reading events are left behind. Products movement can thus be precisely recorded.

One of the important technological advances that targets large-scale traceability is the so-called "Networked RFID" [4]. The basic idea behind the Networked RFID is to use the Internet to connect otherwise isolated RFID systems and software. Networked RFID not only eases the integration of distinct RFID systems, but more importantly, addresses the limitations of passive tags (e.g., communication, computation, and storage) [3]. 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. The EPCglobal Network—designed by the Auto-ID Labs and developed further by the EPCglobal[2]—is a recent notable effort for Networked RFID. The EPCglobal Network is an architecture to realize a "data-on-network" system, where

[1]http://www.ozfoodnet.org.au.

[2]http://www.epcglobalinc.org.

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.150

RFID tags contain an unambiguous ID and other data pertaining to the objects are stored and accessed over the Internet.

However, RFID traceability is not a single-layer problem. Large-scale global RFID networks have the potential to generate unprecedented amounts of data related to individual objects. An important challenge centers on the efficient management and sharing of this data with traceability applications [5], [3], [6]. A system architecture for data gathering, processing and sharing must also be scalable in order to deal with the data collected from networked RFID systems. For efficient processing and storage, data models must be carefully designed. To allow business users to make useful decisions and analysis we must support different types of traceability queries, perhaps also by exploiting some high-level business logic.

The aim of the work reported in this paper is to enhance the fundamental understanding of RFID traceability networks and to develop a novel approach for efficient and effective sharing of RFID data in large-scale, distributed traceability environments. Our main contributions are summarized as follows:

– We have analyzed a wide range of traceability applications and abstracted a generic reference framework of RFID traceability networks. A classification of different types of traceability queries has also been proposed after a careful analysis of essential characteristics of traceability applications.
– We proposed a novel data model for RFID traceability networks that eliminates the data dependencies between organizations. This data model serves as a global schema that allows the formulation of queries without knowledge on how the data is stored, where it is located, and how the tracking queries are executed.
– We proposed a framework for scalable RFID traceability networks, where an RFID tracking engine enables peer-to-peer RFID data sharing across multiple organizations.

The remainder of the paper is organized as follows. Section II gives a brief overview of the related work. Section III presents a generic framework and a classification of queries for RFID traceability applications. Section IV and Section V describe our traceability data model and P2P traceability framework, respectively. Finally, Section VI reports the implementation and some preliminary performance study, and Section VII provides some concluding remarks.

## II. Related Work

In spite of many small, closed-loop RFID pilot applications (e.g., Sydney Harbour Bridge Toll Collection), there remain significant challenges in developing and deploying large-scale and distributed RFID applications (e.g., nationwide supply chain management across companies), as highlighted by MIT researchers in [5]. Central to realizing such a traceability network is the management of RFID data, particularly on how to share RFID data that are typically collected by individual companies across the network. Centralized solutions are clearly not feasible for large-scale applications because of the unique characteristics of RFID data such as streaming and large volume [7], [3], [8].

In the past few years, some research work on RFID data management has emerged. Bornhovd et al. [9] give an overview of their existing RFID infrastructure. Under "Lessons Learned", they emphasise that companies need to overcome their reluctance to collaborate because the full potential of RFID technology can only be unlocked through collaboration and data sharing across sites and organizations. Hu et al. [10] present a new bitmap data type for ORACLE DBMS to support RFID-based item tracking applications. Wang et al. [7] propose a temporal data model for their RFID data management system. An RFID cube is introduced in [11] to support warehousing and analysis of massive RFID data sets. Unfortunately, none of these papers presents solutions to the challenges imposed by independent organizations sharing data in large-scale and distributed applications.

The most notable proposal is the EPCglobal Network, which consists of a network with nodes, and a number of central registries that the nodes can utilize. Each node offers a simple, standardized query interface, called information service, to a repository with RFID data. An application can use the standardized query interface of a repository in order to obtain data. A Discovery Service is proposed to keep track of all object movement between nodes. Whenever an object moves from one node to another, the information is published to the Discovery Service. The fact that all nodes have to continuously publish data to the centralized Discovery Service implies the same limitation as the centralized approaches we mentioned above.

## III. RFID Traceability Networks and Queries

RFID enabled traceability networks have many important applications. There are different ways to implement these traceability applications, which may differ in hardwares, system architectures, scope, and geographical characteristics. However, the core component of all these traceability applications can be abstracted as a *network of nodes* from which information related to objects (e.g., movements from one location to another, travel time) can be extracted. After carefully analyzing a wide range of traceability applications, we propose a generic reference framework (Figure 1) that is agnostic to various traceability applications by modeling elements essential for traceabil-

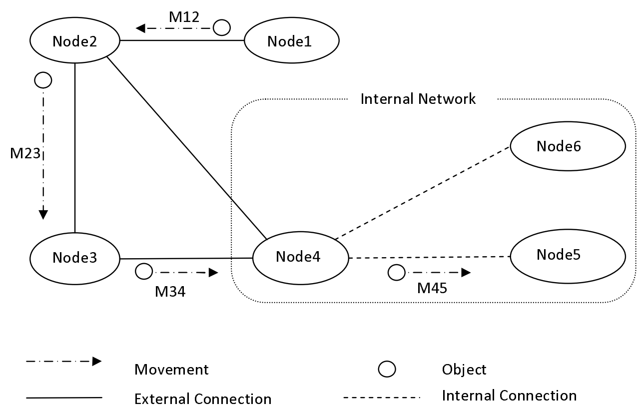



**Figure 1. Reference Framework**

ity. In this section, we first introduce the generic reference framework. We then classify different types of queries that support traceability applications.

## A. The Reference Framework

The generic reference framework (Figure 1) consists of the following components:

- *Node.* Nodes represent observation points in a traceable RFID Network. A node can be a geographic location of an organization or an internal location within an organization. Physically, each node may represent an RFID reader antenna installation to collect and forward RFID data associated with *objects* passing through its detection area. It should be noted that not all the physical locations with RFID reader(s) may be formulated as a node in the traceability network. The number of nodes will vary with user requirements and the location granularity level. For example, in a supply chain, the internal flow of goods inside a distribution center may not be of any interest to other trading partners while it may be critical for the distribution center to manage its inventory. As a result, for the partners, observation points in the distribution center do not count as *nodes*, while for the distribution center, they do.
- *Object.* An object represents a tagged item with a globally unique identifier.
- *Connection.* A connection is a link between nodes. It is established statically (for example, by the partnership of organizations). However, it is *quasi-static* because these partnerships or supply paths may change over time. Each node may be characterized by several properties or meta data (e.g., distance to neighboring nodes, possible methods and cost of travel).
- *Network.* A network is a set of composite *connections* that is quasi-static. It represents the direct or indirect relationship between *nodes*. According to data sharing policies, networks are categorized into two types, *Open-Loop* networks and *Closed-Loop* networks. Within a Closed-Loop network, data is shared by nodes that belong to the same organization. On the other hand, nodes in an Open-Loop network normally belong to different organizations.

The following lists several concepts that encapsulate the dynamic relationship of objects in a traceability network.

- *Movement.* This captures the movement of an object from a source node ($\mathcal{N}_s$) to a destination node ($\mathcal{N}_d$). A movement can be presented by the triplet {$\mathcal{N}_s$, $\mathcal{N}_d$, $\mathcal{T}$}, while $\mathcal{T}$ represents the time taken for the movement.
- *Dwell.* The time an object remains at a node.
- *Path.* A set of ordered movements establishes a path (e.g., {$\mathcal{M}_{12}$, $\mathcal{M}_{23}$, $\mathcal{M}_{34}$} in Figure 1) through the network. Paths are records about the history of an object in both spatial and temporal dimensions.
- *Containment.* Objects may be organized hierarchically. A parent object can contain one or more child objects. This relationship is known as *containment* in our discussion. The containment of objects may be changed between movements. Child objects may be separated from a container (i.e., the parent object) at some point or some objects may join a container. These containment relationship changes must be managed carefully in order to be able to respond to traceability queries. Containment is modeled by the tree structure shown in Figure 2.

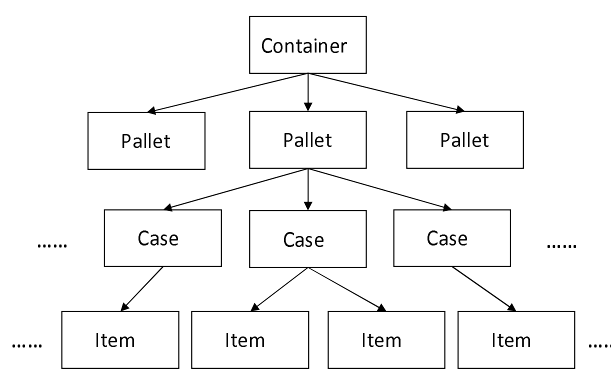



**Figure 2. Containment Tree**

## B. Traceability Queries

The objective of traceable RFID networks is to enable the discovery of past, present and possibly future information of objects. Extracting information for supporting traceability applications requires interrogating traceable RFID networks. Although specific queries may be application dependent, they can be categorized as the following:

**Spatial Queries.** Spatial queries support the retrieval of spatial information about objects within a network such as finding the location of a given object at a specified time or determining places where the object has been for a particular time period. Some typical examples are:

Q1: Where is object $\mathcal{O}_x$ *now*?
Q2: Where was object $\mathcal{O}_x$ on *12th, Oct. 2009*?
Q3: Where was object $\mathcal{O}_x$ last seen?
Q4: Where has object $\mathcal{O}_x$ been between *12th, Oct. 2009* and *20th, Nov. 2009*?
Q5: Where has object $\mathcal{O}_x$ been *last month*?

**Temporal Queries.** In contrast, temporal queries retrieve time-based information about objects such as discovering the time an object was observed at a specific location. Some typical examples are:

Q6: When was object $\mathcal{O}_x$ seen at node $\mathcal{N}_y$?
Q7: How long did object $\mathcal{O}_x$ dwell at node $\mathcal{N}_y$?
Q8: How long did object $\mathcal{O}_x$ take to move from node $\mathcal{N}_x$ to $\mathcal{N}_y$?

**Path Queries.** These queries are designed to discover a part of or the whole life history of an object such as obtaining movement information. Some typical examples are:

Q9: What nodes did object $\mathcal{O}_x$ travel through before it reached node $\mathcal{N}_y$?
Q10: Find the travel path for object $\mathcal{O}_x$ in the last month.
Q11: Find the travel path for object $\mathcal{O}_x$ before it reached node $\mathcal{N}_y$.
Q12: Find the nodes visited by object $\mathcal{O}_x$ after node $\mathcal{N}_y$.

**Statistical Queries.** Statistical queries are designed for data analysis applications (e.g., the applications relying on predictions or estimations related to movements of objects). For example, in a supply chain, it might be interesting to know the average time an object spends in storage at a particular distribution center. Some typical examples are:

Q13: How many objects have been sent from node $\mathcal{N}_x$ to node $\mathcal{N}_y$ last year?
Q14: Which node sent node $\mathcal{N}_x$ the maximum number of objects last year?
Q15: Which node received the minimum number of objects from node $\mathcal{N}_x$ last year?
Q16: What is the average dwell time of object at node $\mathcal{N}_x$?
Q17: What is the total number of objects seen at node $\mathcal{N}_x$ last year?
Q18: What is the expected arrival time for object $\mathcal{O}_x$ at node $\mathcal{N}_x$?
Q19: What is the probability that object $\mathcal{O}_x$ will arrive at node $\mathcal{N}_x$ in the next hour?

**Containment Queries.** These queries aim at extracting containment relationships between objects such as finding the containment relationship of objects at a specified time or node. Some typical examples are:

Q20: What objects were contained in object $\mathcal{O}_x$ on *12th, Oct. 2009*?
Q21: What was the container for object $\mathcal{O}_x$ when it was at node $\mathcal{N}_x$?
Q22: Where was object $\mathcal{O}_x$ packed into object $\mathcal{O}_y$?
Q23: When was object $\mathcal{O}_x$ unpacked from object $\mathcal{O}_y$?

**Metadata Queries.** Traceability networks have domain specific information captured as properties or meta data, which are associated with the nodes of the networks or the objects themselves. Some examples of such data include valid travel paths to other nodes, unit cost of a movement and node description. These kinds of data are very important to traceability applications. Extracting meta data is straightforward in the sense that i) if the meta data is part of the object data model then this information is readily available, and ii) if this meta data is part of a node's property we can retrieve this information directly from a specific node. Some typical examples are:

Q24: What is the cost for object $\mathcal{O}_x$ to move from node $\mathcal{N}_y$ to node $\mathcal{N}_z$?

## IV. An RFID Traceability Data Model

Although RFID data processing has attracted a huge interest in last few years [11], [7], [12], [13], very few of them consider the sharing of RFID data in large-sale and distributed environments [2]. For example, the data model presented in [7] introduces data dependencies between organizations and is not suitable to be applied in a fully distributed environment. EPCglobal Network is a recent notable effort for using RFID in large-scale and distributed environments. Its discovery service, a core component for enabling traceability, is unfortunately a centralized design.

To realize a scalable traceability network in distributed environments, it is necessary to come up a novel data model that eliminates any data dependencies between organizations. This data model will serve as a global schema that allows the formulation of a query without knowledge on how the data is stored, where it is located, and how a tracking query is executed.

By considering the generic reference framework that we have identified in Section III, we propose a conceptual data model for RFID traceability networks. Figure 3 shows the Entity-Relationship (ER) model of the data model, which is ready to be converted to any relational data model. We abstract essential elements of object movement in a traceability network as a set of *entities* and *relationships* between these entities. The fundamental entities include:

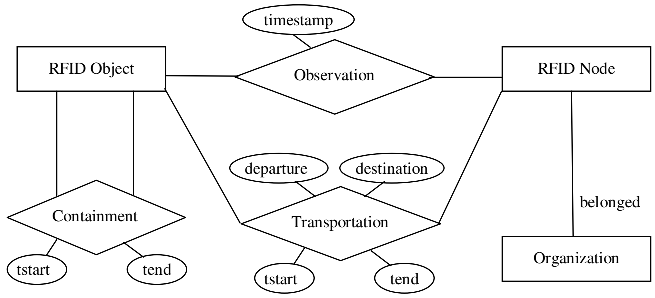



**Figure 3. The ER model for RFID traceability networks**

- *RFID objects* are RFID-tagged objects such as individual items, cases, pallets, and trucks. Objects are uniquely identified using schemes such as electronic product code (EPC).
- *RFID nodes* represent locations equipped with RFID readers. Such nodes can use radio-frequency signals to communicate with RFID tags and read the data stored in the tags.
- *Organizations* can be manufacturers, hospitals, distributors, retailers etc. As mentioned in Section III-A, each RFID node belongs to a particular organization, which can have multiple RFID nodes.

We abstract three important relationships in RFID traceability networks, as the following:

- *Containment* captures the containment relationship (i.e., parent-child) between objects. For example, a case is packed into a pallet at a particular RFID node. This relationship has two attributes `tstart` and `tend`, indicating the lifespan of the relationship. It should be noted that there might be multiple levels of hierarchy in the containment relationship.
- *Observation* is generated when an object interacts with an RFID reader. This relationship has one attribute `timestamp` indicating the time point when the interaction (i.e., reading) occurs.
- *Transportation* captures the movement of an RFID object to an RFID node, which is important in traceability networks. This relationship has four attributes. `tstart` and `tend` indicate starting and ending times of the movement, while `departure` and `destination` indicate the leaving and arriving RFID nodes, respectively. We call these two properties the *information of object path* (IOP).

The proposed transportation relationship is generic to model all the dynamic relationships that we identified in Section III-A. For example, a *dwell* relationship (i.e., the time an object remains at a particular node) can be modeled by setting the `departure` and `destination` as the same node. Abstracting IOP attributes in transportation relationship is extremely important to improve the performance of a distributed traceability network. With IOP, it is possible to minimize the number of nodes to be visited without flooding the queries to all nodes in the network. The second part of this task is to develop an approach for acquiring IOP. A naive approach for obtaining IOP is to extract such information from other enterprise data such as ordering and billing. Obviously, this approach not only requires high synchronization of RFID data and enterprise data, but is impractical for many applications where such enterprise data is unavailable (e.g., traffic tracking). In this work, we provide a generic approach in Section V.

## V. Peer-to-Peer RFID Traceability Networks

Manipulating traceability queries typically needs to deal with data from multiple sites. For example, for the query "What nodes did object $\mathcal{O}_x$ travel through before it reached node $\mathcal{N}_y$" (see Q9 in Section III-B), we have to execute it at each organization where there is an entry with $o_{id}$=$\mathcal{O}_x$ in the respective partition of the relation *transportation*[3].

Existing approaches such as distributed and federated database systems deal with query processing across multiple sites. However, the global knowledge about the data distribution is a priori. In addition, P2P content sharing networks have been extensively researched in the last few years [14]. Such approaches either flood queries or use globally available information about data distribution (e.g., in the form of distributed hash-table [15]) to route queries.

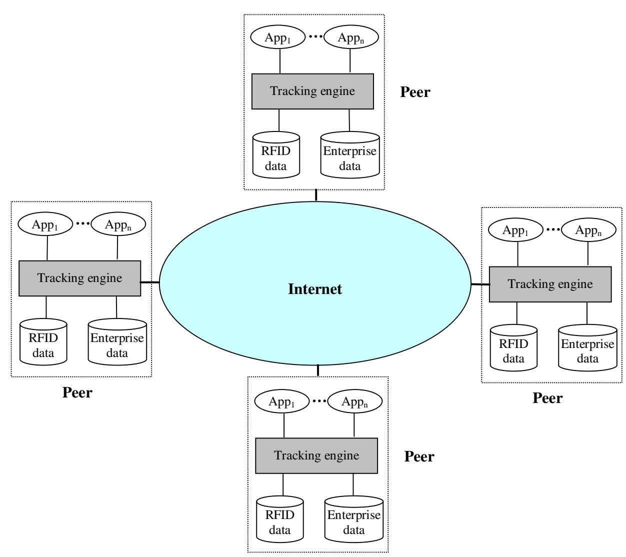



**Figure 4. Peer to Peer traceability networks**

[3] $o_{id}$ is an attribute of the transportation relationship that represents the id of the RFID object.

Central to our approach towards scalable RFID traceability network is an RFID *tracking engine* that supports peer to peer (P2P) RFID data sharing across multiple organizations. As shown in Figure 4, each organization is viewed as an equal peer of the network. Within the peer, applications can access local RFID data, as well as remote RFID data of other peers using the tracking engine via the Internet. Our system implements the tracking engine using Web service technology to address critical issues of large-scale, distributed applications such as flexibility, heterogeneity, and openness.

The fundamental principle of our approach is to process a query locally to the extent possible and, if necessary, enhance it using locally available information before forwarding it to appropriate remote organizations for further processing. The algorithm is given in Figure 5. Given a global traceability query, our query engine rewrites it exploiting local data, locates remote data sources, forwards it to the query engines of remote peers, and finally combines local and remote results. This process repeats itself at the other nodes. It should be noted that due to the introduction of `departure` and `destination` of the relationship `transportation` in our data model, it is possible to trace the relevant organizations without flooding the query to all nodes in the network. It is also worth noting that our algorithm only deals with queries that involve data from multiple organizations. It is quite straightforward to deal with queries that can be answered in a single organization such as Q6, Q7, Q13, Q16, Q17, and Q21 (see Section III-B).

As discussed in Section IV, the IOP (i.e., `departure` and `destination` attributes) plays an important role in the distributed query processing. However, in most cases, this information is not directly available. In this work, we propose a P2P algorithm (see Figure 6) for acquiring IOP by indexing the latest location of an object. The general idea is that we find a deterministic node as the *gateway* node for an object. This node is determined by the *lookup* operation of DHT [14]. The latest location of the object is indexed (stored) on that gateway node. Each time when we query this object, its latest location can be extracted from the *gateway* node. In addition, the `departure` and `destination` nodes' IOPs are updated by the gateway node (lines 4 to 9 and lines 11, 12). The gateway node updates the information every time the object arrives at a new node (line 10). As a result, the IOP information is connected naturally forming a distributed double-linked list.

We assume that all organizations have the knowledge of the proposed data model when dealing with global traceability queries. However, organizations are not required to use the data model in their implementations. If this is the case, a mapping between our data model and the local schema of an organization should be sufficient for the rewriting of global queries to conform to the local schema.

**Input**: a global traceability query $q$
**Output**: the set of query result $\mathcal{R}$

Let $\mathcal{T}$ be the transportation table of the peer where $q$ is initiated.
Let $\mathcal{T}_{dp}$ and $\mathcal{T}_{dt}$ be the departure and destination peers of $\mathcal{T}$.
$\mathcal{R} \leftarrow \phi$
$\mathcal{R} \leftarrow$ executeQuery($q$) /*execute $q$ locally*/
**while** $\mathcal{T}_{dp}$ is not null **do**
    forward $q$ to $\mathcal{T}_{dp}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup$ executeQuery($q$) /*execute $q$ remotely*/
    set $\mathcal{T}_{dp}$ to the departure peer of $\mathcal{T}_{dp}$
**end while**
$\mathcal{R} \leftarrow$ postprocess($\mathcal{R}$)
**return** $\mathcal{R}$

**Figure 5. Algorithm for P2P query processing**

**Input**: an object $\mathcal{O}$ arrived at node $\mathcal{N}$
**Output**: the $departure$ and $destination$ attributes (IOP)

1: let $\mathcal{N}_{gateway}$ be the result of DHT lookup for $\mathcal{O}$.
2: $\mathcal{N}$ sends a message containing $\mathcal{O}.id$ to $\mathcal{N}_{gateway}$.
3: $\mathcal{N}_{gateway}$ receives the message, looks up index for $\mathcal{O}$ as $index$.
4: let $last\_location$ be $index.last\_location$.
4: **if** $last\_location$ is null
5:     sends a message to $\mathcal{N}$, indicates that $\mathcal{N}$ is the starting point for $\mathcal{O}$.
6: **else**
7:     sends a message to $\mathcal{N}$: $\mathcal{O}$ is from $last\_location$.
8:     sends a message to $last\_location$: $\mathcal{O}$ has arrived at $\mathcal{N}$.
9: **end if**
10: $\mathcal{N}_{gateway}$ sets $index.last\_location$ as $\mathcal{N}$.
11: $\mathcal{N}$ receives the message from $\mathcal{N}_gateway$, sets the $departure$ of $\mathcal{O}$ as $last\_location$.
12: $last\_location$ receives the message from $\mathcal{N}_gateway$, sets the $destination$ of $\mathcal{O}$ as $\mathcal{N}$.

**Figure 6. Algorithm for IOP Acquisition**

# VI. Implementation and Performance Study

The idea presented in this paper has been realized in a large research project, PeerTrack[4], that aims at developing a comprehensive platform for developing scalable RFID traceability networks. In particular, we implemented the tracking engine (see Figure 4) based on JXTA[5], a language and platform independent protocol for P2P communication. Each organization is considered as a JXTA peer equipped with a local RFID database and the algorithm for processing traceability queries. Currently, our system can handle most of the queries identified in Section III-B such as pedigree extraction, movement monitoring, and containment queries. Some statistical queries involve estimation and predication techniques (e.g., Q18 and Q19), which will be the target of our future work. Figure 7 shows the result of a pedigree query using our system.

To validate the feasibility of our approach, we also implemented a medicine distribution supply chain, which

[4]http://www.cs.adelaide.edu.au/peertrack
[5]http://jxta.dev.java.net.

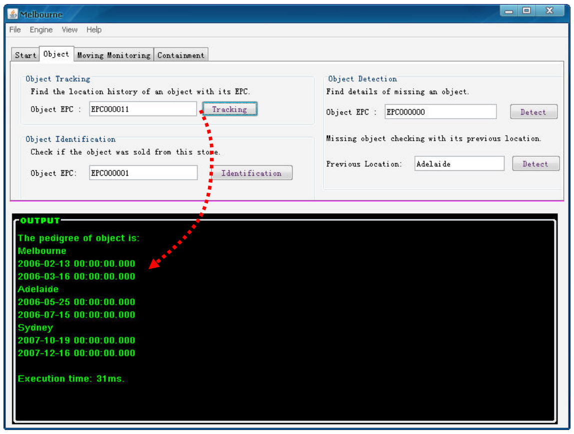


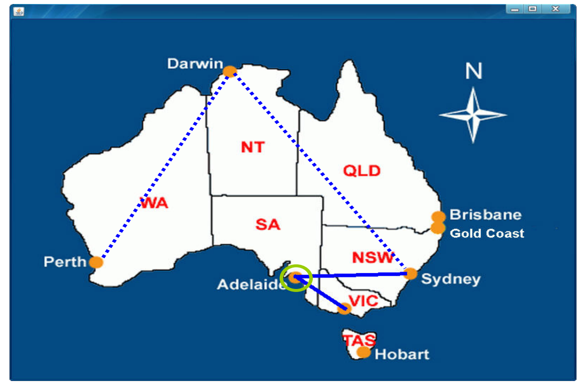

**Figure 7. Traceability queries processing**

involves major cities in Australia. Figure 8 shows some screenshots from the application that monitors the movement of a product. In particular, Figure 8 (a) depicts the movement of the product, which starts from Adelaide in a pallet, arrives at Sydney, is traveling to Darwin and then Perth in a case (dotted lines). Figure 8 (b) shows the results from an object tracking query using our system (i.e., the detailed temporal information displayed on the map).

**Performance Study.** We have conducted some preliminary performance study with emphasis on evaluating the performance of our proposed approach. Due to the space constraints, we report here the result of one of our experiments.

The purpose of this experiment is to examine two approaches' performance when the number of nodes in the network increases, i.e. the scalability. We simulated a dynamic supply chain network with various number of nodes in one machine with a configuration of Intel Core 2 Quad 2.4GHz CPU and 3Gb RAM. All nodes are implemented following JXTA standards and are separate processes, thus they communicate with each other through JXTA P2P network. Each node runs two query engines: the one with flooding queries and our P2P tracking engine. In the experiment, we considered two cases: i) flooding the queries to all nodes, and ii) our P2P query processing. We simulated a simple program that submitted a set of pedigree queries, which involves multiple nodes (the number of nodes involved was randomly generated), simultaneously to both engines for processing. We then compared the query processing time and the query processing traffic. The number of nodes ranges from 10 to 50. We ran the same set of 10 queries on both engines and counted the total time used and the number of messages involved.

Figure 9 (a) gives the average query processing time of the two engines. We can see that the processing time of our approach remains steady while the flooding approach takes more time when the number of nodes in the network




(a)

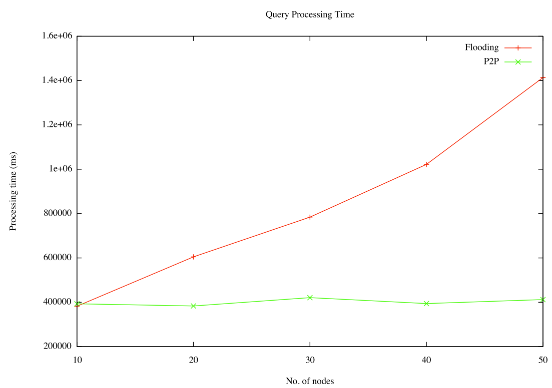

(b)

**Figure 8. The traceability network of a medicine supply chain: (a) product movement, (b) traceability query results**

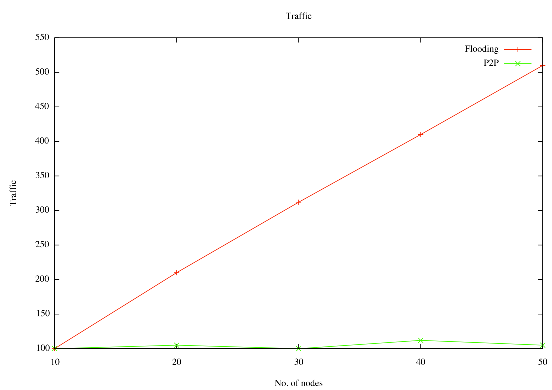
increases. Figure 9(b) shows the average query processing traffic. Similarly, flooding approach consumes much more traffic than our approach. From the results we can conclude that, our approach is scalable when the network expands. The query processing time is only relevant to the nodes visited during the object's lifetime. On the other hand, flooding approach has to ask every node in the network in order to find out whether the object has visited. It has the complexity of $O(N)$ where $N$ is the number of nodes in the network, which is not scalable.

## VII. Conclusion

The ability of RFID technology for precisely identifying objects at low-cost and without line of sight creates many new and exciting application areas. In the future, this wide range of applications will make RFID an integral part of our daily lives. However, applying RFID presents many challenges on information systems and technologies such as scalability. In this paper, we have presented a generic reference framework and a classification of queries of traceability applications. We further proposed a novel RFID data model and a P2P traceability architecture, which




(a)




(b)

**Figure 9. Performance study: (a) query processing time, (b) query processing traffic**

lay the foundation for developing scalable traceability applications in large-scale and distributed environments. The ideas presented in this paper have been validated in the implementation and some preliminary experimental studies.

We view our work presented in this paper as a first step towards realizing scalable RFID traceability networks. Our ongoing work includes extension of the RFID tracking engine to support estimation and predication queries using statistics and probability technologies. Another important plan is to build more traceability applications on top of the architecture and conduct further detailed performance study of the approach in a WAN environment.

## References


[1] G. Borriello, "RFID: Tagging the World," *Communications of the ACM*, vol. 48, no. 9, pp. 34–37, September 2005.

[2] G. Roussos, S. S. Duri, and C. W. Thompson, "RFID Meets The Internet," *IEEE Internet Computing*, vol. 13, no. 1, pp. 11–13, January/Feruary 2009.

[3] Q. Z. Sheng, X. Li, and S. Zeadally, "Enabling Next-Generation RFID Applications: Solutions and Challenges," *IEEE Computer*, vol. 41, no. 9, pp. 21–28, September 2008.

[4] G. Roussos, *Networked RFID: Systems, Software and Services*. Springer, 2008.

[5] S. S. Chawathe, V. Krishnamurthy, S. Ramachandran, and S. Sarma, "Managing RFID Data," in *Proc. of the 30th International Conference on Very Large Data Bases (VLDB'04)*, Toronto, Canada, September 2004.

[6] F. Wang, S. Liu, and P. Liu, "Complex RFID Event Processing," *The VLDB Journal*, vol. 18, no. 4, pp. 913–931, 2009.

[7] F. Wang and P. Liu, "Temporal Management of RFID Data," in *Proc. of the 31st International Conference on Very Large Data Bases (VLDB'05)*, Trondheim, Norway, September 2005.

[8] M. J. Franklin, S. R. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, and W. Hong, "Design Considerations for High Fan-in Systems: The HiFi Approach," in *Proc. of the Second Biennial Conference on Innovative Data Systems Research (CIDR'05)*, Asilomar, CA, USA, January 2005.

[9] C. Bornhövd, T. Lin, S. Haller, and J. Schaper, "Integrating Automatic Data Acquisition with Business Processes Experiences with SAP's Auto-ID Infrastructure," in *Proc. of the 30th International Conference on Very Large Data Bases (VLDB'04)*, Toronto, Canada, September 2004.

[10] Y. Hu, S. Sundara, T. Chorma, and J. Srinivasan, "Supporting RFID-Based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype," in *Proc. of the 31st International Conference on Very Large Data Bases (VLDB'05)*, Trondheim, Norway, September 2005.

[11] H. Gonzalez, J. Han, and X. Li, "FlowCube: Constructing RFID FlowCubes for Multi-dimensional Analysis of Commodity Flows," in *Proc. of the 32nd International Conference on Very Large Data Bases (VLDB'06)*, Seoul, Korea, September 2006.

[12] Y. Diao, B. Li, A. Liu, L. Peng, C. Sutton, T. Tran, and M. Zink, "Capturing Data Uncertainty in High-Volume Stream Processing," in *Pro. of the 4th Biennial Conference on Innovative Data Systems (CIDR'09)*, Asilomar, CA, USA, January 2009.

[13] Y. Bai, F. Wang, P. Liu, C. Zaniolo, and S. Liu, "RFID Data Processing with a Data Stream Query Language," in *Pro. of the 23rd International Conference on Data Engineering (ICDE'07)*, Istanbul, Turkey, April 2007.

[14] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[15] B. T. Loo, J. M. Hellerstein, R. Huebsch, S. Shenker, and I. Stoica, "Enhancing P2P File-Sharing with an Internet-Scale Query Processor," in *Proc. of the 30th International Conference on Very Large Data Bases (VLDB'04)*, Toronto, Canada, September 2004.