Cost-effective Product Traceability System based on Widely Distributed Databases

WAKAYAMA Shirou
shirou.wakayama@toshiba.co.jp
DOI Yusuke and OZAKI Satoshi and INOUE Atsushi
ydoi@isl.rdc.toshiba.co.jp, satoshi.ozaki@toshiba.co.jp, inoue@isl.rdc.toshiba.co.jp
Corporate Research & Development Center, Toshiba Corporation

***Abstract*— A cost-effective and easy way of introducing a product traceability system is to start from a small system and gradually extend it to a large-scale system. Traceability systems used in previous field tests are unsuitable for large-scale deployment because they use a single, centralized database. This paper describes an extendable traceability system proposed by Toshiba that employs distributed databases. Our proposed system called LoTR consists of two components, a DHT-based ID-DB resolution mechanism and TraceBack method. The ID-DB resolution mechanism is used to acquire a DB location from an ID. We apply the DHT to accomplish required scalability. We have developed a DHT-DNS mounter to achieve seamless access to DHT-based ID-DB resolution from ordinary DNS resolvers. In addition, the TraceBack method creates links to bind distributed databases. We have evaluated the operation of the system with a real field trial. As the next step, we shall analyze its qualitative behavior through large-scale experiments.**

***Index Terms*— RFID, Product Traceability, DHT, DNS, Distributed DBs**

I. INTRODUCTION

In response to concerns regarding the safety of food, many feasibility studies have been conducted on product traceability systems to trace how food has been produced, processed and transferred. Although a small system is adequate for a study whose scope is limited, the system will become larger as it is deployed step by step to include many shops, and fine granularity of property management will be required. On the other hand, it is economically unattractive to construct a traceability system if the desired scalability potential cannot be achieved without a large initial system. Hence, a traceability system architecture is required that can initially be small in scale and subsequently be extended gradually.

We propose a scalable traceability system that can start from a small scale. We constructed a prototype traceability system to confirm its effectiveness. We also discuss smooth and gradual growth of the proposed system.

In this paper, we present the structure of existing traceability systems. Analyzing issues regarding data and database distribution, we argue that proper distribution of the database is important as the scale of the system increases. Finally, we describe our proposed system and its characteristics.

II. PRODUCT TRACEABILITY

A. Mechanisms of Product Traceability

Product traceability systems are systems for ascertaining where a product was before coming to the current location (trace back) and where a product went from its source (trace forward). Among the diverse applications of systems of this kind are food traceability, e.g. for beef, and product recalls of home electronics equipment.

To provide traceability, a system must be able to identify products item by item. In addition, it must be able to record product data on each item. For example, a customer consuming beef may desire various types of information on feed, growth of the animal, temperature during storage and distribution, and processing, in addition to the identity of the animal. In this paper, we call such product-related data "*ID-property*." Clearly, barcodes used in current distribution systems lack the capacity to record such a large amount of information; some types of RFID tags can however record this information. However, much memory is necessary to store ID-properties in the RFID tag, so that the tags become expensive. Not only suppliers but also consumers are reluctant to accept a 5-dollar RFID tag on a 5-dollar pack of beef to ensure food safety.

Hence, some proposals on traceability systems envisage the storage of ID-properties on a networked database of data linked to items, and attachment of product ID numbers only to the items.

Since EPCglobal[1] uses a 96-bit ID, conventional barcodes cannot carry sufficient information to express an ID. Two-dimensional codes or simple and cheap RFID tags can store at least 1Kbit of static data, which is sufficient for an ID.

B. Existing Traceability Systems

In the pig production chain, many countries have their own identification systems[2]. For example, France and EU countries have laws concerning the keeping of livestock, animal breeding and identification (Regulation of June 1993 and Circular 4021, 29 April 1997 for France, 64/432/CEE; 92/102/CEE, 27 November 1992; 97/12/CEE, 17 March 1997 for the EU, etc.). According to these laws, farms maintain a logbook and the movements are recorded using an identifier put in ear tags or other technology. A governmental organization is

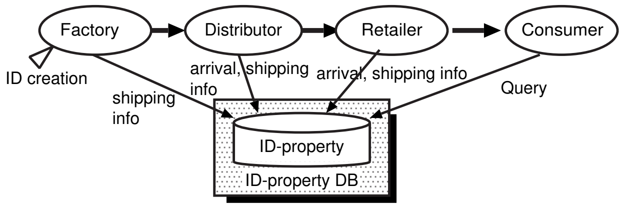

Figure 1. System used in field test of Gillette Company: It has only one DB that stores all ID-property. A user sends the query to acquire ID-property with the ID as the key.

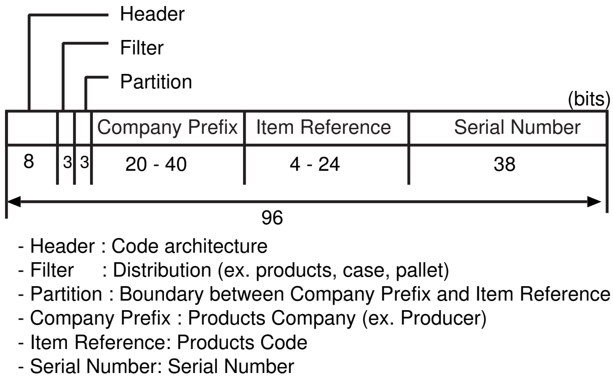

Figure 2. Example of electronic product code (EPC) structure (SGTIN-96): EPC global creates the ID resolution mechanism to map this ID structure to the tree structure.

responsible for assigning the official numbers as identifiers. For example, "FR29051020" is assigned to a farm, and "91325" is assigned to a sow. These numbers and pig movements are maintained in a certain department, and although there are no official rules covering what happens after the animal leaves the farm, many processing companies attach an individual slaughter number to each carcass.

Sainsbury's trial[3][4] does not attempt item-level traceability but transport unit-level traceability, preferably using recyclable transport containers. Because tag costs constitute the majority of the RFID system implementation costs, it is important to reduce the number of tags, and so this trial focused on transport unit-level traceability.

However, tag costs are falling fast. For example, on May 1, 2006 SmartCode announced the first ever $0.05 tag[5].

Although there are many examples of RFID traceability research or trials, such as the Wal-Mart trial[6], Tesco [7], and the METRO Group Future Store Initiative[8], they tend to be focused on the RFID tag itself, and there are few reports on traceability systems. We focused on a traceability system with the attributes of scalability and extendibility.

Most of the existing traceability systems, such as that used in the Gillette Company field test in 2003, have only one database that stores every ID-property in the system (Fig. 1). To read an ID-property, clients directly send a query with the ID to the database and get the ID-property.

With this simple architecture, management and development are very easy. However, as the system grows, the capacity must be increased. For example, a DB may store data from hundreds of farms and tens of thousands of stores. Although the problem may be latent in the initial phase of deployment, if the system eventually handles greater amounts of products or additional categories of products such as pork, vegetables or home electronics, it is difficult for a single database to handle all the data and queries. We expect costs to rise in the case of this architecture even if the database is expanded using clustering techniques.

C. Existing Proposals using Multiple Networked DB

Models proposed by standardization groups such as EPCglobal use a mechanism to find the database corresponding to an ID. We call this mechanism "*ID-DB resolution*".

Such a model can distribute databases using manufacturer code or item class code. Each database may be as small as a cheap commodity computer, and the total cost of constructing a larger-scale system may be lower than in the case of the single-database model.

On the other hand, because an ID-DB resolution mechanism must be used for every query, the mechanism (called Object Name Service: ONS[9], in EPCglobal) may be a bottleneck. According to version 1.0 of ONS specification, DNS (Domain Name Systems) are relied on to handle huge numbers of queries to preclude bottlenecks.

This proposal assumes the ID code format has the same structure as SGTIN-96(Figure 2), which is one example of the ID structures used in the EPCglobal system. In SGTIN-96, a company prefix part and an item reference part are mapped in a tree structure as in the case of a conventional domain name. However, the serial number part still has a lot of space without any structure and DNS assumes that an ID has a domain-based structures.

A cache should be effective in a domain-based structure because it is has a tree structure. However, DNS could not achieve sufficient performance for its unstructured part because the cache function is ineffective. Thus, it is difficult to realize an ID-DB resolution mechanism based on DNS.

III. Proposed Traceability System

A. Database Management per Management Domain

We call the fundamental economic unit of a traceability system a "*management domain*." For example, farms, slaughterhouses, warehousing, distributors, markets, and consumers are candidate management domains for a beef traceability system. A traceability system is a system that records and utilizes the ID-properties of products. An ID-property includes the effects on a product of environment, processing, alteration within a management domain, and transfer of the product between management domains.

From a management perspective, it is preferable to have an ID-property database for each management domain because the data of any given domain can be managed and controlled and its secrets can be secured. A database may manage private data in addition to traceability data using product IDs as keys. We call this database for an individual management domain which has traceability a TDB (Traceability DataBase).

This kind of system may work as a distribution and supply chain management system as well, since this used the same equipment as a traceability system.

To accomplish this, it is necessary to solve two problems. First, the system must be able to realize an ID-DB resolution mechanism with serial number granularity. For this purpose, a different ID-DB resolution mechanism is required. We propose an ID-DB resolution mechanism adopting a distributed data management method called DHT (Distributed Hash Table).

The other issue is how to retrieve ID-properties from distributed DBs. ID-properties are placed in the DB along with product distribution information. We try to solve this issue by using a bidirectional link creation mechanism called TraceBack.

The TraceBack mechanism is similar to the TrackBack mechanism widely used in weblog systems.

B. ID-DB Resolution Mechanism Based on DHT

DHT is a data management framework organizing and looking up data through a table, consisting of many networked computers. Each computer behaves as a node with the same algorithm, the nodes sharing the service load autonomously. Many algorithms have been proposed for DHT, including Chord[10] and SkipNet[11]. DHT uses the hash value of a key (product ID in this case) to distribute data among nodes. This approach is similar to the lack of structure in the serial number part of product IDs.

CoDoNS [12] is a name service using a prefix-matching DHT called the Beehive proactive cache framework. After CoDoNS has run for 6 hours to adjust the extent of proactive caching, CoDoNS can retrieve information with lower latency than the conventional DNS server. In addition, CoDoNS reduces the vulnerability of CoDoNS to denial of service attacks.

On the other hand, DNS has been widely accepted as the interface of the ID-DB resolution mechanism for various applications. It is undesirable to mandate a different interface for every application of our new ID-DB resolution mechanism. Such requirement would create a great difference between the pure DNS-based ID-DB resolution mechanism (ONS) and our approach. To ensure the widespread acceptance of the DHT-based ID-DB resolution mechanism, the framework must be integrated with a DNS interface. A simple protocol translator may work but this creates a performance bottleneck (Fig. 3).

We propose a DHT-DNS mounter[13] specified according to DNS name delegation. Figure 4 shows this mechanism. The DHT-DNS mounter can manage a DNS query and a DHT query. An ordinary DNS query is recursive. In the tree structure part of the ID, the query moves recursively and reaches one DHT-DNS mounter. That DHT-DNS mounter takes over that DNS query and sends it to DHT nodes as a DHT query. In other words, DHT is used in processing the structure-less part of the ID. Since the DHT-DNS mounter can process DNS and DHT queries, the approach mitigates the bottleneck between DNS and DHT without any modification of the DNS interface or applications of the mechanism.

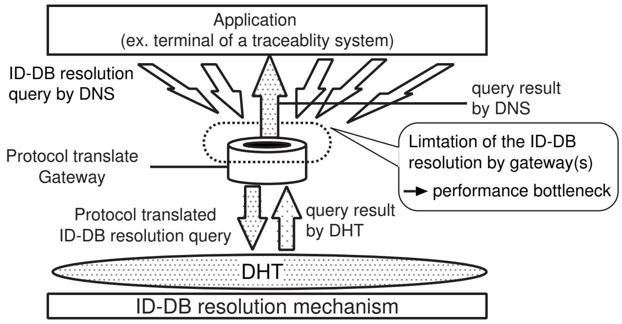

Figure 3. DHT-DNS protocol conversion gateway bottleneck: Only a few gateways and many queries create a bottleneck.

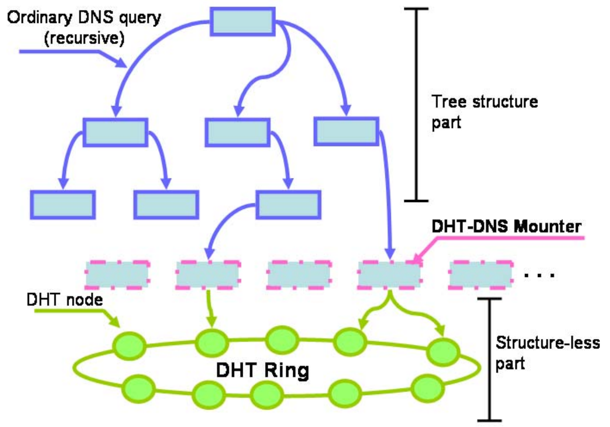

Figure 4. DHT-DNS Mounter: Use DNS in the tree structure part, DHT in the structure-less part. Many DHT-DNS mounters can convert the queries.

C. Inter-Database Cooperation using TraceBack

A web document system called weblog is equipped with the TrackBack mechanism to create links among articles. We proposed a similar mechanism called *"TraceBack"* for traceability systems with an independent TDB for each management domain. TraceBack links TDBs as items are transferred among the management domains corresponding to those TDBs. Figure 5 is a schematic chart of TraceBack operations.

TDB records item arrival and shipment. For every arrival and shipment, TDB records *links*. A link is information indicating the source TDB of an arrived item or the destination TDB of a forwarded item. As the TDB records a link, it sends notification called "TraceBackPing" to the TDB at the other end of the link (usually the source of the item). The TDB on the other end adds a reference link (usually the destination of the item) according to the received TraceBackPing. As a result, a bidirectional link is created between the source-destination pair of TDBs in the respective management domains. Users can trace items among management domains using this bidirectional link.

A TDB can create links with other TDBs by using the TraceBack method. Users can trace TDBs following the links. However, since an ID does not have the information of which TDB has stored that ID, users must first find TDB from the ID by using a conversion mechanism, namely the ID-DB resolution mechanism.

The ID-DB resolution mechanism stores the last TDB location in the supply chain. That is, users can find the last TDB location from an ID by using ID-DB resolution mechanism and then find all the TDBs in the supply chain by following the link from each successive TDB. In addition, users can also acquire the ID-properties of that product by asking every TDB.

Therefore, when the product reaches the management domain of a TDB, the TDB sends a TraceBackPing to the previous TDB, whose location can be obtained by the ID-DB resolution mechanism. After the TraceBackPing is sent, the TDB updates the ID-DB resolution mechanism. In other words, the TDB registers its location with the ID-DB resolution mechanism. Since the ID-DB resolution mechanism stores only the last TDB location, the information to be stored by the ID-DB resolution mechanism is not so large. However to provide redundancy, the ID-DB resolution mechanism may store the history of the TDB locations.

Figure 5 shows the following flow of these messages.

1) When the management domain B receives the product, the TDB B in the Management domain B receives the ID of that product using a RFID reader or some other method.
2) The TDB B sends a query to the ID-DB resolution mechanism to get the previous TDB location.
3) The ID-DB resolution mechanism sends back the last TDB location. Here, it is the TDB A.
4) The TDB B sends notification to the TDB A and creates a link to the TDB A. The TDB A receives the notification and creates a link to the TDB B.
5) The TDB B updates the location of the product in the ID-DB resolution mechanism.

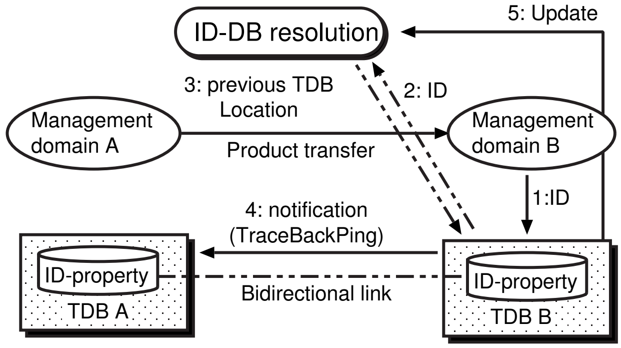

Figure 5. TraceBack mechanism for traceability systems: The TDB sends notification to the previous TDB that can be found by ID-DB resolution mechanism.

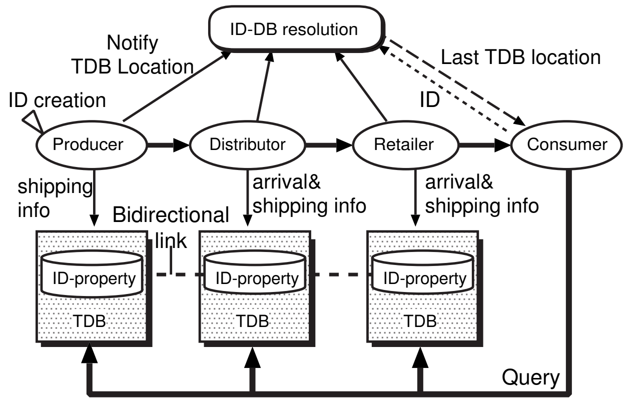

Figure 6. Proposed method of database distribution using TraceBack and ID-DB resolution mechanism: The DB that stored an ID-property most recently can be found by using a DHT-based ID-DB resolution mechanism. The other TDBs can be found by following the links.

We call a product traceability system with such a link structure and ID-DB resolution mechanism, an *LoTR: Linked Object TRaceability system*. Figure 6 shows the structure of the LoTR. Using the LoTR, we can add databases easily thus enlarging the product traceability chain. Characteristics of LoTR are:

1) Link: There is a bi-directional link between the DBs. A link should have the network addresses of the two DBs, timestamps, etc.

A bi-directional link for one ID should have two destinations; downstream and upstream. Without both, we cannot provide a bi-directional traceability function. Furthermore, sometimes a link has no value. For example, manufacturers and consumers have a link in only one destination.

There are several identification formats for a network address. We choose the URL specification[14] of the network location because URL is familiar in current Internet architecture.

TABLE I.
LINK INFORMATION

name	description	example
id	Identifier	123456789A
next_link	URL of next TDB	http://slaughterhouse.example.jp
prev_link	URL of previous TDB	http://farm.example.jp
in_date	inbound date	2006-02-12 10:12:43
out_date	outbound date	2006-02-14 15:48:01

Basic messages related to a link are of two types: TraceBackPing and LinkQuery.

A TraceBackPing message requests a TDB to create a link for an ID. After receiving a TraceBackPing message, the TDB creates a link indicating the address of the TDB that sent the message.

A LinkQuery message requests link information stored in the TDB. Not only a TDB but also the consumer can send a LinkQuery message to a TDB. Because the TDB does not manage queries recursively, one LinkQuery message can retrieve only one pair of links, the next link and the previous link. Thus, a message sender must send plural LinkQuery messages to get all link information for one ID. This specification increases the sender's workload, but decreases the TDB workload.

2) Divide and construct: Many products are in turn composed of several products. For example, a hamburger is made from bread, hamburger meat, lettuce, etc. In addition, many products are divisible into several products. The traceability system must also execute these division and construction processes.

We can solve this problem by creating new ID(s) when products are divided or constructed. As the TDB knows the relation between the old ID(s) and the new ID(s), the TDB can respond with that relation information when the TDB receives a LinkQuery message. Since the message sender can get information about old ID(s) in the response, the sender may send a LinkQuery message about old ID(s) and can get a link.

Figure 7 shows the tracing mechanism after a product is divided. In this figure, product X is divided into product Y and product Z at TDB_N.

In this case, TDB_N knows that product X was divided into Y and Z and transferred to TDB_O and TDB_P. When TDB_N receives a LinkQuery message about Y or Z, TDB_N responds with link information about X. From this response, the message sender can trace product Y or Z as product X.

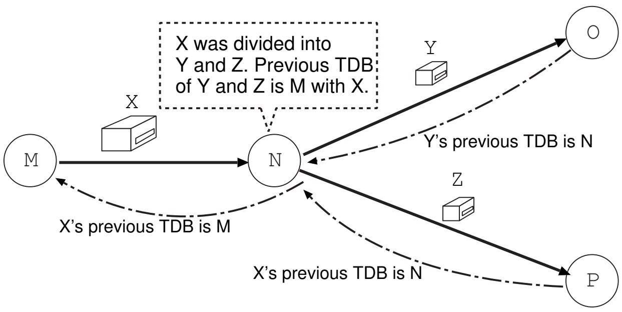

Figure 7. Tracing mechanism after a product is divided: The TDB_N knows all information about product X.

IV. IMPLEMENTATION

With the DHT-based ID-DB resolution mechanism and the inter-database cooperation mechanism using Trace-Back, the traceability system LoTR can manage an ID-property database in each management domain. To confirm that our proposed method works correctly, we have implemented a prototype product traceability system.

A number of recent events (epizootics of swine fever, BSE (Bovine Spongiform Encephalopathy) and FMD (Foot and Mouth Disease)) have engendered anxiety concerning food safety among the public. In Japan, beef is the product about which there is the greatest anxiety concerning food safety. Hence, we designed a test of a beef traceability system in February 2006. In this section, we describe the experiment.

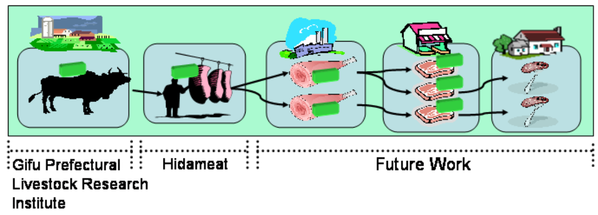

Figure 8. Field overview: Although there are many management domains in one supply chain, only two management domains are incorporated in this experiment.

A. Field Overview

The experiment was supported by the following organizations:

- Gifu Prefectural Livestock Research Institute,
- Hidameat Agricultural Cooperatives.

Gifu Prefectural Livestock Research Institute is a farm and Hidameat Agricultural Cooperatives is a slaughterhouse. Cattle fattened on the farm are sent to JA Hidameat where they are slaughtered, and each carcass is divided into two sides of beef. These sides of beef are sold at auction and distributed bearing the "Hidagyuu" brand after being held in a cold storage at the slaughterhouse for about 3 days.

Figure 8 shows this flow. The real supply chain continues after the slaughterhouse. These parts will be considered in future work.

In this experiment, the farm manages information on each animal. The following values are ID-properties of the farm:

- name,
- date of birth,
- sex,
- breed,
- maternal breeding,
- paternal breeding,
- weight history by month,
- temperature history by month,
- fodder,
- drug usage history.

An animal swallows an RFID tag with a temperature sensor. Using the tag, the temperature in the stomach can be measured from outside. Figure 9 shows an animal on a scale equipped with an RFID reader. Its weight and temperature can be measured simultaneously.

In the slaughterhouse, we collected the surface temperature of each side of beef in the cold store every 10 minutes. A user can get these values via a user interface.

B. System Overview

Figure 10 shows an overview of the experimental system.

Figure 9. The scale with RFID reader: The RFID reader is attached to the side of scale. The temperature and the weight of the animal can be measured simultaneously.

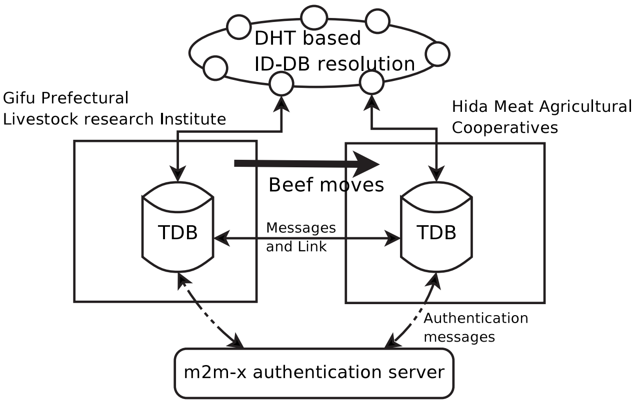

Figure 10. System overview of the experiment: There are two TDBs in the each management domain and one DHT based ID-DB resolution mechanism in the Internet. Each TDB communicates the m2m-x authentication server to authenticate each other.

There are two TDBs; one for the farm and one for the slaughterhouse. These TDBs have traceability information and ID-properties.

We implemented the Chord algorithm in our ID-DB resolution mechanism using Python, and 10 DHT nodes were used.

Figure 10 also shows the security mechanism. This security mechanism is a part of the automatic connection mechanism called UOPF[15], and the implemented function is called m2m-x. It can automatically authenticate and encrypt a connection between TDBs. By using m2m-x, TDB spoofing or a man-in-the-middle attack can be avoided.

C. Functions

Functions provided by the experimental prototype are as follows.

Providing traceability information: A TDB provides traceability information to users via user interfaces. A user can easily acquire information such as the place of production, the logistics path, and the place where sales was made to a consumer.

This implementation can trace returned products. For example, even if an animal goes from the farm to the slaughterhouse, returns to the farm, and then goes back to the slaughterhouse again, these product movements are traceable.

Providing ID-property: As described in IV-A, TDBs at the farm and the slaughterhouse store many ID-properties. A user can get these ID-properties via a user interface. User interfaces visualize ID-properties in simple formats such as graphs.

V. Discussions

In this section, we describe the pros and cons of the proposed system. First, a DHT-based ID-DB resolution mechanism has the following advantages.

1) It can comprise from one node initially and grow gradually and smoothly as the service load increases.
2) It can reduce its scale as the service load decreases.
3) It has no bottleneck in theory; the service load is shared evenly between nodes.
4) It binds data on a node automatically.

On the other hand, it has the following disadvantages.

1) It is difficult to execute a process that manages only a part of an ID.
2) There is no way to handle blocks of continuous IDs.

These disadvantages are inherent to the system due to the hash functions in DHT. However, since an item-level traceability system does not have these special processes, we expect our system not to have these disadvantages.

Second, a LoTR system has the following advantages.

1) The size of a management domain is independent of total system scale.
2) It can start from a small domain (such as an in-store management system) and grow gradually and smoothly.

On the other hand, users of a LoTR system must send queries to many databases to collect all the data of an item. This specification may cause a bottleneck.

The two components of this proposed traceability system allow the system to start on a small scale and grow gradually and smoothly. As the next step in our research, we intend to perform quantitative evaluation of a large-scale network through analysis and experiments.

A. Discussion of the Experiment

In the above experiment, we confirmed traceability using the proposed system in a real supply chain between a farm and a slaughterhouse. We acquired temperature from the sensor, and fodder and weight history as ID-properties, and provided them to the user using visualization aids such as graphs.

We encountered the following problems:

– Because the slaughterhouse does not have a network infrastructure, we used a wireless network of cellular phones (IMT-2000). As a result, there was a large latency.
– We use the DHT via the Internet, but still it is required to set the TDB in the field. However, the fields did not have a such space to set the machines,and it was necessary to stack the machines. This may cause a heat problem.

B. Miscellaneous problems

1) Link disconnect: If a TDB is temporarily or permanently down, the user cannot trace using that link. However, if one database has multiple links to every mth database in the sequence, the user can trace. Each TDB operator can decide what m should be, but m should be over 3.

In addition, TDB may have links to TDBs whose length is far greater than odd numbers such as $m = 3, 5, 7$. This method can reduce the number of TDBs whose links must be stored.

2) Mismatch: Mismatch between the last location of a product that the ID-DB resolution mechanism has stored and the real current location may occur even when a TDB updates the location that the ID-DB resolution mechanism stores. There are thought to be two possible reasons.

– Disorder of the update message,
– Undeliverability of the update message.

Only the TDB that received an ID from an RFID reader sends an update message to ID-DB resolution mechanism. If the TDB maintains message numbers and responses properly, the disorder in the message sent will not occur. If the TDB resends messages when ID-DB resolution mechanism does not respond, undelivered messages can be largely avoided.

3) Link loop: When a product is defective or for some other reason, it can happen that the same product returns to the management domain it was sent from, such as a farm in our experiment, and then sent to a slaughterhouse etc. again.

A product may come back to a management domain more than once for some reason. For example, a cow at a slaughterhouse must be returned to the farm if it has fever. In this case, since the same ID appears, if TDB cannot classify differently it, the user may not be able to trace the product.

The solution to this link loop problem is for every TDB to store an ID each time with a different serial number that is unique in that TDB. By using this serial number, TDB can recognize that the same ID has been entered multitude times. The serial number may be a timestamp. However, because the returning operation may happen twice or more in one day, the serial number should be a timestamp containing a second order number or another unique number.

C. Security and Privacy

In this mechanism, any TDB can create a uni-directional link in that TDB without sending TraceBack-Ping message. As a result, it is possible for an attack to occur that causes the user to trace to the wrong link or to a malicious link. An attack of this type can be avoided by having the two TDBs mutually confirm the link. In addition, if TDBs exchange their signatures by a certain method, each TDB sends signed TraceBackPing and the receiving TDB can verify the validity of the sending TDB. Thus, attack becomes more difficult.

Additionally, the TDB may also have a link to the next-next TDB in the supply chain. This additional link allows a TDB to verify the TDB in two directions: previous TDB and next TDB.

The URL is the only identification of the TDB. A malicious TDB can disguise itself as another TDB by hijacking the URL. If the disguise is successful, the malicious TDB can attack any TDB or links for example, by falsifying a link, leading to the wrong supply chain or providing the wrong ID-property. In the experiment, we prevent spoofing by using the m2m-x security mechanism to authenticate the communication.

As described in III-C.2, this system can monitor division and combination of the product. However, only one TDB has this information, and this becomes a single-point-of-failure.

Privacy concerns: It is essential to ensure consumer privacy in the case of item-level traceability. Our proposed system describes traceability through to the retailer, not through to the consumer. When real operation begins, the retailer should deactivate the tag after the consumer purchases RFID-attached products. In addition, RFID tags should be clearly marked so that all consumers understand RFID tags are attached. EPCglobal produce guidelines for protection of consumer privacy[16].

VI. Conclusion

RFID tags are expected to come into widespread use in supply chain management (SCM) systems together with two-dimensional codes. We expect high demand for product traceability systems, in which an item has its own ID through out the supply chain.

We propose a low-cost product traceability system with the attributes of scalability and extendibility. We think this system configuration will facilitate wider deployment of RFID systems, particularly large-scale, inter-domain deployment. Furthermore, we believe that proof of the effectiveness of the proposed system will facilitate RFID deployment.

As the next step, we intend to perform quantitative evaluation in order to confirm the scalability of the proposed ID-DB resolution mechanism.

An example of such research is measurement of response time according to the amount of entries and load variation in a test bed such as the StarBed [17] system. Characteristics between a DNS-based mechanism and a DHT-based mechanism will be different, and we expect DHT to have better scalability.

Since this experiment has only two fields, it is insufficient to measure the scalability of the proposed system. We shall increase the number of fields. In addition, we will consider measuring the LoTR scalability in a virtual large-scale supply chain using the StarBed system.

ACKNOWLEDGMENT

This research is supported/funded by the Ministry of Internal Affairs and Communications of Japan.

REFERENCES

[1] EPCglobal, http://www.epcglobalinc.org/
[2] F. Madec et al. Traceability in the pig production chain, ev. sci. tech. Off. int. Epiz., 2001, 20 (2), 523-537
[3] How Sainsbury's Transformed Its Supply Chain, Supply Chain Management Review May 7, 2004
[4] RFID in the grocery supply chain ? A remedy for logistics problems or mere hype?, ECR Student Award Paper 2002.
[5] SMARTCODETM CORP. announces the world's first 5 cent RFID tag http://www.smartcodecorp.com/newsroom/01-05-06.asp
[6] Wal-Mart EPC Pilot Begins Transition to Implementation, http://walmartstores.com/GlobalWMStoresWeb/ navigate.do?catg=25&contId=4802
[7] Collins, J. (2004) Tesco Begins RFID Rollout. RFID Journal, September 28, 2004.
[8] The Future Store Initiative: A platform for the future of retailing, http://www.future-store.org/
[9] Michael Mealling. Auto-id Object Name Service(ONS)1.0. Auto-ID Center Working Draft, August 2003.
[10] Stoica, I. et al. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In Proc. ACM SIGCOMM, 2001, p.149 – 160.
[11] Harvey, N. et al. Skipnet: A scalable overlay network with practical locality properties. In Proc. the 4 th USENIX Symposium on Internet Technologies and Systems, 2003, http://www.usenix.org /events/usits03/tech/harvey.html.
[12] Venugopalan Ramasubramanian and Emin Gun Sirer. The design and implementation of a next generation name service for the internet. In ACM SIGCOMM, September 2004.
[13] Yusuke DOI. DNS meets DHT: Treating Massive ID Resolution Using DNS Over DHT. In Proceedings of The 2005 International Symposium on Applications and the Internet(SAINT 2005), p.9 – 15.
[14] Uniform Resource Identifier (URI): Generic Syntax, RFC 3986.
[15] Ubiquitous Open Platform Forum, http://uopf.org/en/
[16] Guidelines on EPC for Consumer Products, http://www.epcglobalinc.org/public_policy/ public_policy_guidelines.html
[17] Toshiyuki, M. et al. Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed, Tridentcom 2005, pp.274–282.

WAKAYAMA Shirou recieved a MS in media and governance from the Keio University, Japan, in 2003.

He is currently a researcher at Corporate Research and Development Center, Toshiba Corporation, in Kawasaki, Japan. His current research interests include RFID technologies, ubiquitous networking, sensor and context-aware networking.

Yusuke DOI is a researcher at Corporate Research and Development Center, Toshiba Corporation in Kawasaki, Japan. His research interests include distributed systems, naming architectures, and privacy on RFID systems. Mr. Doi is a member of the ACM and the IEEE Computer Society.

Satoshi OZAKI received a PhD in electrical and computer engineering from the Yokohama National University, Japan, in 1995.

He is currently a Research Scientist at Corporate Research and Development Center, Toshiba Corporation, Japan. His current research interests include ubiquitous networking, Internet services and network security of embedded systems.

Dr. Ozaki is a member of the IEEE Computer Society and the IEICE.

Atsushi Inoue graduated the department of Mathmatical engineering and Information Physics, the University of Tokyo, Japan, in 1985.

He is currently a Chief Research Scientist at Corporate Research and Development Center, Toshiba Corporation, Japan. His current research interests include Internet protocols, particularly IPv6, IP mobility and network security.

Mr. Inoue is a member of the ACM and the Information processing Society of Japan.