PAPER *IEICE/IEEE Joint Special Section on Autonomous Decentralized Systems Technologies and Their Application to Networked Systems*

Data Management for Large-Scale Position-Tracking Systems

Fumiaki INOUE[†*], **Yongbing ZHANG**[†a)], ***Nonmembers*, *and*** **Yusheng JI**[††], ***Member***

SUMMARY We propose a distributed data management approach in this paper for a large-scale position-tracking system composed of multiple small systems based on wireless tag technologies such as RFID and Wi-Fi tags. Each of these small systems is called a *domain,* and a domain server manages the position data of the users belonging to its managing domain and also to the other domains but temporarily residing in its domain. The domain servers collaborate with each other to globally manage the position data, realizing the global position tracking. Several domains can be further grouped to form a larger domain, called a *higher-domain*, so that the whole system is constructed in a hierarchical structure. We implemented the proposed approach in an experimental environment, and conducted a performance evaluation on the proposed approach and compared it with an existing approach wherein a central server is used to manage the position data of all the users. The results showed that the position data processing load is distributed among the domain servers and the traffic for position data transmission over the backbone network can be significantly restrained.

key words: *position tracking, position data management, wireless tag technology, large-scale system*

1. Introduction

Tracking the positions of mobile users and navigating their destinations are two issues that have long been identified as key components in emerging mobile-computing applications [4], [7], [14], [15], [18]. Mobile devices with geographical positioning system (GPS) function are widely used in outdoor position-tracking systems. However, GPS-based position tracking is limited to outdoor environments since GPS is satellite dependent and the satellite signals are usually unavailable most of the time indoors [10], [16]. Several approaches [12], [13], [16], [22] have recently been proposed for indoor position tracking using wireless tags using Wi-Fi [3] or radio frequency identifier (RFID) [19] functionality. The position data in such a system are collected at the user sites and then transferred to the central server of the system via the network such as the Internet and the Intranet. Most current position-tracking systems using wireless tags have been developed by various vendors using distinct wireless technologies, and furthermore a single vendor may even use a specific platform that differs from others for each application. Therefore, the position-tracking service is generally limited in size, resulting in a high construction cost and low availability. Due to this lack of scalability, if a large-scale system is constructed based on a long haul network using the current approach, a large amount of network bandwidth must be wasted for position data transmission when users are far away from the central server. Furthermore, the central server may be overloaded with position data if a number of users move around in the system.

We propose a distributed data management approach in this paper that combines multiple small position-tracking systems into a large-scale system. The service area realized by the proposed approach is therefore expanded to the entire area covered by all the smaller systems. Each small system is similar to those used in previous researches and is called a *domain* in this paper. A user belongs permanently to one domain, called his/her *home-domain,* but he/she can freely travel in not only his/her home domain but also any other domain within the system. The domains are interconnected with one another directly in a flat topology or in a hierarchical topology. Each domain server manages the position data of the users belonging to its managing domain and also to the other domains but temporarily residing within its domain. The home-domain server of a user stores all the position data of the user while the user resides in its domain and can also obtain the position data of the user from other domain servers if necessary. The advantages of the proposed approach are that the processing load of the position data is distributed amongst the domain servers in the entire system and that the traffic for position data transmission over the backbone network can be limited. To our knowledge, current research in the literature does not address this issue.

The outline of this paper is as follows. Section 2 reviews the related work. Section 3 describes the unified data format in the proposed approach and explains the proposed data management approach in detail. Section 4 shows the implementation details of the proposed approach in an experimental environment. Section 5 presents the performance evaluation of the proposed approach. Section 6 concludes the paper.

2. Related Work

The most successful position-tracking applications nowadays are those developed for outdoor environments on mobile cellular phones with a GPS function [1], [2], [11], [17]. However, the GPS is generally unavailable indoors since the

Manuscript received April 5, 2010.
Manuscript revised August 26, 2010.

[†]The authors are with the Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305-8573 Japan.
[††]The author is with the National Institute of Informatics, Tokyo, 101-8430 Japan.
[*]Presently, with the Mobile Network Division, Softbank Mobile Corp.
a) E-mail: ybzhang@sk.tsukuba.ac.jp

DOI: 10.1587/transcom.E94.B.45

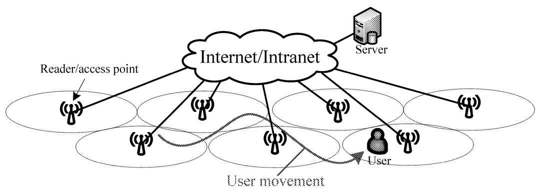

Fig. 1 Traditional position-tracking system.

GPS signals from satellites are blocked by buildings or other obstacles. The position of a cellular phone user can also be estimated using the signals received from nearby base stations but the estimation error is large. Furthermore, a key problem with this kind of application is that it is difficult to trace all the mobile users in real time since their position data should be stored and processed within the central server of the mobile phone service provider. It is also difficult to obtain the location-specific information around the current position such as the floor and the room information.

Wireless devices with infrared, active RFID and Wi-Fi tags are expected to be good choices for position tracking both indoors and outdoors and recently many researches have explored position tracking using wireless tag technologies [12], [13], [15], [16]. The service (coverage) area of a position-tracking system using wireless tags is composed of the areas covered by all the readers or the access points deployed within the system that can sense the signals from the wireless tags as shown in Fig. 1. Readers/access-points are permanently deployed without mobility and used as the position markers, and the position of a user with a tag is determined by the reader/access-point to which the user's tag is connecting or by the estimation result based on the signals sensed by the readers/access-points around the user.

Infrared badges are used by Olivetti Research Laboratory to develop an indoor position tracking system [22]. The limitation of this system is the line-of-sight requirement and the short-range signal transmission. Recently, wireless devices with active RFID tags [19] or Wi-Fi tags [3] are expected to play a key role in indoor position-tracking applications [8], [12], [13], [16], [20], [21]. However, a common problem with these systems is that they are limited in size and are homogeneous in specification. That is, each system is basically developed on a specific architecture and is not compatible with others, thus yielding a low scalability. Therefore, it is difficult to combine the position-tracking services of two such systems into one even when the two systems use the same technology.

Panasonic proposed an RFID-tag based system [13], [21] for tracking the positions of children enrolled in a single school on their way to the school. Each student carries an active RFID tag and the readers are equipped on the electric poles along the roads within the school zone. When a reader senses a student with a RFID tag passing through its coverage area, it sends the tag information to the central server that is maintained at the school. The parents of each student can access the central server via the Internet to check on the current or historical positions of their children. Since each school with such a system operates on its own position-tracking service, the position of a child cannot be traced if he/she moves out of his/her own school zone and enters another school zone even when the visiting school uses the same technology. Therefore, the service availability is extremely low and the construction cost is inevitably high.

All of the researches in [5]–[7], [12], [16], [22] focused on the problem of position estimation using RFID or Wi-Fi tags. The system architectures using those approaches are all the same in the sense that they are homogeneous and the position data of the users are all transferred via the network, such as the Internet or the Intranet, to the system central server. Some approaches have proposed using reference tags [12] or the signals from neighboring tags [16] to improve the accuracy of this position estimation. The main problem with these approaches is again that the network size is limited even though it is not explicitly described in some of these referenced papers.

Some applications using wireless tags were developed for large-scale systems used in the airfreight or shipping industries such as air cargo container tracing [20] and sea container tracing [8]. A logistics company can provide position-tracking services covering a wide area across different countries or oceans using this technology to trace the current positions of containers and to optimize the container scheduling strategies. The position data of the tracing targets are all sent to the company's central server via a long haul communication network. Since the current position of a container may be far away from the central server, the data transmission should pass through a long path to the central server. Therefore, the position data transmission from the current position of the container to the central server wastes a large amount of network bandwidth. Furthermore, when the number of tracing targets increases, the data processing load at the central server may be excessive.

3. Proposed Approach for Position-Data Management

We propose a distributed data management approach in this paper that combines multiple small systems (domains) to form a large-scale system. Each domain corresponds to a stand-alone system in the previous researches. Each user belongs permanently to one domain, called his/her *home domain,* as users in previous systems. However, each user can autonomously move not only in his/her home domain but also in any other domain, appropriately called a *foreign domain*, in the system. The home-domain server of a user receives the position data of the user directly from the readers/access-points deployed in its domain, and furthermore, can also obtain the position data from the foreign-domain servers if the user has ever been in those domains. Each domain server also stores the position data of the users belonging to other domains but having temporarily resided in its domain. In order to obtain the most current or historical positioning of a user, one can directly access the user's home-domain server or via a Web interface.

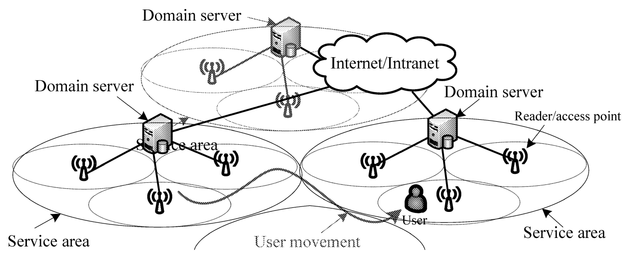

Fig. 2 Flat-domain topology.

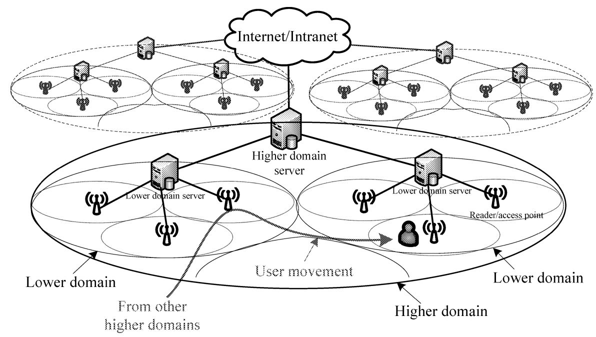

Fig. 3 Hierarchical-domain topology.

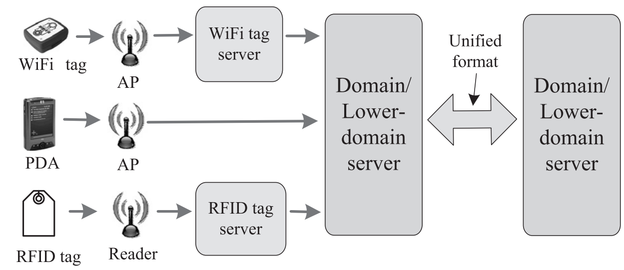

Fig. 4 Unified position data.

Tag identification	
Domain id	Tag id

(a) Tag identification in flat topology

Tag identification		
Higher domain id	Lower domain id	Tag id

(b) Tag identification in hierarchical topology

Fig. 5 Tag-identification information.

The proposed network architectures in this paper can be classified into either *flat* or *hierarchical* topologies as shown in Figs. 2 and 3. In a flat topology, the domain servers are directly connected to one another possibly via the Internet or an Intranet, as shown in Fig. 2. In a hierarchical topology, on the other hand, several domains, which are specially called the *lower-domains* here are grouped to form a larger domain, called a *higher-domain*, as shown in Fig. 3. The higher-domain servers are usually interconnected via the Internet. For simplicity, we sometimes use the term *domain* to mean either a domain in a flat topology or a lower-domain in a hierarchical topology. A domain/lower-domain is composed of all the areas covered by the readers/access-points deployed within the domain and in other words denotes the entire area covered by the readers/access-points directly connecting to the domain server. A higher-domain on the other hand is composed of all the lower-domains under itself. Furthermore, the area covered by a reader indicates the area that the reader can sense.

Since position-tracking applications using Wi-Fi and RFID tags can be similarly implemented, in this paper we only show the implementation for the position tracking using RFID tags for sake of simplicity. The readers deployed within a domain for sensing the signals from the RFID tags are connected directly or via a specific tag server to the domain server depending on the tag-sensed data processing mechanism, as shown in Fig. 4. For example, in this paper we used the RFID tags made by RF Code Inc. [19], and the sensed tag data can be sent to the domain server only via the tag server, which is called the concentrator. On the other hand, more smart wireless devices like personal digital assistants (PDAs) can directly transfer data to any domain server. We propose using a unified data format for data management since the domains may use various distinct RFID tags and their data formats may have no compatibility. The advantage of the unified format is that the differences of both the hardware and software between various systems can be absorbed at the domain level and the position data can be processed in the same way throughout the system.

A reader and a domain server have their own identifiers. When a user enters a domain, the reader(s) sensing the user send(s) the user information, say, the tag ID, together with its/their own identifier(s) to its domain server. The domain server then records this fact for the user together with a timestamp showing the event time instant. Note that multiple readers usually sense a user at the same time, and therefore, the position data records at the user home-domain server may be overlapped. These data can be used for estimating a more accurate position for the user using an approach in [5], [6], [12], [16]. When a user is away from his/her home domain, his/her current position information is transferred by the server of the current residing domain back to his/her home domain server.

3.1 User and Position Identifications

Each RFID tag has a unique identifier and its identification data format is shown in Figs. 5(a) and 5(b). The tag identification data include the serial number and the domain identifier to which the tag belongs. Furthermore, in a hierarchical topology, a tag identification data additionally includes the identifier of the home higher-domain to which the tag's user permanently belongs, as shown in Fig. 5(b).

In a flat topology, each domain has a unique identifier and the server of each domain knows the identifiers of all the other domains. In a hierarchical topology, on the other hand, each higher-domain has a system-wise unique identifier while each lower-domain has only a unique iden-

Tag ID	Domain info		Reader info		Connect timestamp	Connect duration
	Name	IP address	Name	Position info		

(a) Position data format in flat topology

Tag ID	High domain info		Low domain info		Reader info		Connect timestamp	Connect duration
	Name	IP address	Name	IP address	Name	Position info		

(b) Position data format in hierarchical topology

Fig. 6 Position-data format.

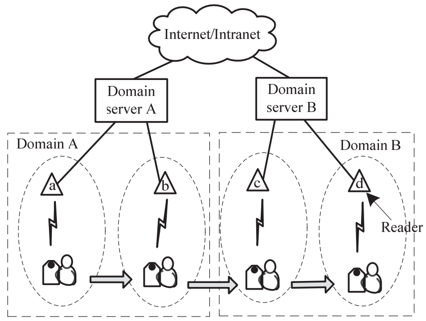

Fig. 7 Position registration in flat topology.

tifier under the corresponding higher-domain. Each lower-domain sever knows the identifiers of both its corresponding higher-domain and the other domains under the same higher-domain. Each higher-domain server knows the identifiers of all the other higher-domains and all the lower-domains under itself.

3.2 Unified Data Format

When a reader senses a user, i.e., the user's tag ID, as shown in Fig. 5, it sends, usually via a specific server dedicated to the readers of the same domain, the user's tag ID together with its own identifier to the domain server in a flat topology and to the lower-domain server in a hierarchical topology, respectively. The domain server collects the position data in different formats and then transforms them into a unified format throughout the entire system.

When a domain server receives the position data of a user residing in its managing domain, it stores the user's tag ID in its database along with the reader ID and the time instant at which the user connected to the reader. A domain or higher-domain server transfers the position data according to the unified data format shown in Fig. 6, but some data items can be left empty. For example, when a user moves to a foreign domain in a flat topology, the foreign-domain server generally does not send the reader ID that sensed the user to the user home-domain server unless the reader ID is clearly requested by the user home-domain server.

3.3 Position Data Management in Flat Topology

In a flat topology, each domain server manages the position data of the users belonging to its managing domain and also to the other domains but temporarily residing in its domain. That is, each domain server knows the exact posi-

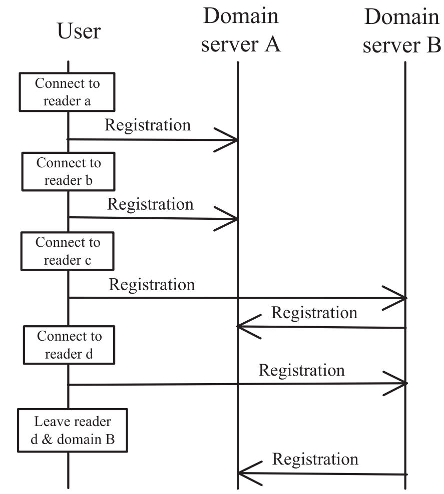

Fig. 8 Registration sequence in flat topology.

tion about a user who is residing in its managing domain such as to which reader the user has ever connected and how long the connection has been sustained. Domain servers are connected with one another by the Internet or an Intranet, as shown in Fig. 7. For a user, all the domains other than his/her home domain are called his/her *foreign domains*. When a user, e.g., domain A's user, enters or leaves a foreign domain, e.g., domain B, a position registration message is sent to his/her home-domain server, i.e., server A. However, if the user moves back and forth between the coverage areas of the readers under the same foreign domain, there is no need to send any position update message to the user home-domain server. The user home-domain server can instead ask a foreign-domain server for detail position information if necessary. Therefore, the position data messages transferred over the network can be reduced since when a user is away from his home domain detail position data may not be always necessary. Furthermore, the processing load for processing position data at the user home-domain server can also be alleviated.

Figure 8 shows the position-registration sequence for a user when the user has travelled in the order shown in Fig. 7. Suppose that the user belongs to domain A and entered its home domain A from the outside. The position registration procedures for the user within its home domain A and a foreign domain B are shown as follows.

A. Position registration in home domain.

When the user entered the coverage area of reader *a* in domain A, reader *a* registered the user to domain server A with the user tag ID, reader *a*'s ID, and the time instant at which the user entered the coverage area of reader *a*. When the user has left the coverage area of reader *a* and then entered the coverage area of reader *b*, reader *b* registered the user to domain server A with the user tag ID, reader *b*'s ID, and the time instant at which the user entered the coverage area of reader *b*. When the user has left the coverage area of reader *a*, reader *a* sent the time length the user was connected to reader *a* to domain server A. When the user has left the coverage area of reader *b*, reader *b* also sent the time length the user was connected to it to domain server A.

B. Position registration in foreign domain.

When the user entered the coverage area of reader *c* in domain B, reader *c* registered the user to domain server B with the user tag ID, reader *c*'s ID, and the time instant at which the user entered the coverage area of reader *c*. If the user entered domain B for the first time, domain server B sends the user tag ID, domain server B's ID, and the time instant at which the user entered domain B to the user home-domain server A. When the user moves back and forth between the coverage areas of different readers in domain B, the position registration is performed only at domain server B, but no update message is sent to the user home-domain server A. When the user finally has left domain B, domain server B sends a position registration message and the time length the user has been in its domain to the user home-domain server A.

As described above, when a user is travelling in its home domain, its home-domain server knows the user tag ID, the reader ID to which the user is connected, and the time duration the user was connected to each reader. On the other hand, when the user is travelling in a foreign domain, the user home-domain server only knows the user tag ID, the foreign-domain ID, and the time duration the user was in the foreign domain. However, the user home-domain server can ask the foreign-domain server for more detail information about the user, e.g., to which readers the user was connected.

3.4 Position Data Management in Hierarchical Topology

In a hierarchical topology, the network is hierarchically constructed wherein the domains in a flat topology are treated as lower-domains and several lower-domains are further grouped as a higher-domain, as shown in Fig. 9. Therefore, the coverage area of a higher-domain is composed of all the lower-domains under the higher-domain. For example, as shown in Fig. 9, higher-domain A covers the areas of lower-domains A1 and A2. Higher-domain server A can communicate directly with any other higher-domain server, e.g., B, and also with the lower-domain servers under itself, e.g., A1 and A2. Furthermore, a lower-domain server under a higher-domain, e.g., A1 under A, can communicate directly with not only its higher-domain server, i.e., A, but also with other higher-domain servers, e.g., B, and additionally with the lower-domain servers under the same higher-domain, e.g., A2. If a lower-domain server, e.g., A1, wants to communicate with another lower-domain server under a different higher-domain, e.g., B1 under B, it should pass through the higher-domain server, i.e., B, of the target lower-domain server, i.e., B1.

Each lower-domain server maintains the position data of the users belonging to its managing domain and also to the other domains but temporarily residing in its domain. On the other hand, a higher-domain server maintains only the position data of the users belonging to the other higher-domains and temporarily residing in its lower-domains. There are three cases in a hierarchical topology for registering a user's position according to where the user is moving around. Suppose that a user belongs to lower-domain A1 of higher-domain A, and the user entered its home lower-domain A1 from the outside and has travelled in the order shown in Fig. 9. The position registration messages transferred between the domain servers are shown in Fig. 10 and the position registration procedures for the user are as follows.

A. Position registration in home lower-domain.

When the user entered the coverage area of reader *a* under lower-domain A1, and then moved to the coverage area of reader *b*, the position registration procedure was the same as that described in "A. position registration in home domain" in Sect. 3.3.

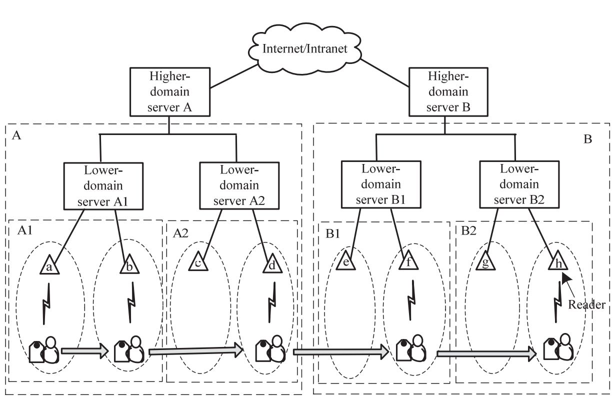

Fig. 9 Position registration in hierarchical topology.

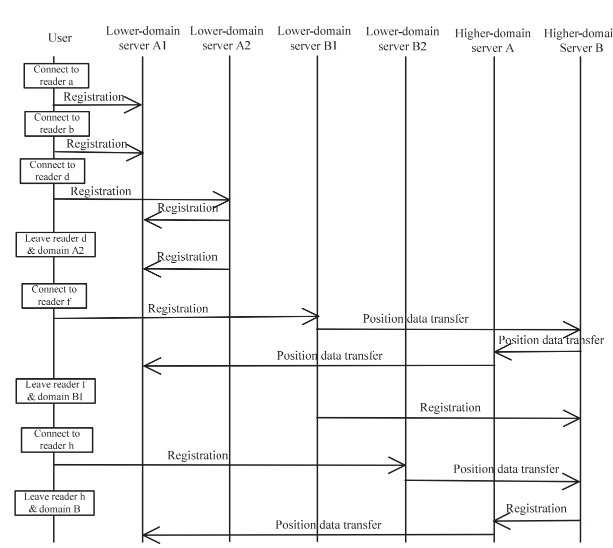

Fig. 10 Registration sequence in hierarchical topology.

B. Position registration in foreign lower-domains under the same higher-domain.

When the user entered lower-domain A2 under higher-domain A and stayed lower-domain A2 or moved back and forth between the coverage areas of readers under lower-domain A2, the position registration procedure was the same as that described in "B. position registration in foreign domain" in Sect. 3.3.

C. Position registration in foreign higher-domains.

When the user entered the coverage area of reader f of lower-domain B1 under higher-domain B, reader f registered the user to lower-domain server B1 with the user tag ID, reader f's ID, and the time instant at which the user entered the coverage area of reader f. Since B1 had no information about the user, it sent the user tag ID, its own ID, and the time instant the user entered lower-domain B1 to its higher-domain server B. When higher-domain server B received the user position data, it identified the home higher-domain of the user, say, A, and then sent the user tag ID, its own identifier, and the time instant at which the user entered the coverage area of higher-domain B to the user higher-domain server A. Then, higher-domain server A forwarded this data to the lower-domain server of the user, A1.

When the user entered the coverage area of reader h in lower-domain B2, reader h registered the user to B2 with the user tag ID, reader h's ID, and the time instant at which the user entered the coverage area of reader h. When the user left domain B1, B1's server sent the user tag ID, its own ID, and the time duration the user was in domain B1 to higher-domain server B. Since higher-domain server B knew the fact that the user has moved from domain B1 to B2 and that both B1 and B2 are under itself, it does not send the position registration message to the higher-domain server of the user, A.

After the user has left lower-domain B2 and then the coverage area of higher-domain B, lower-domain server B2 sent the user tag ID, its own ID, and the time duration the user was in domain B2 to higher-domain server B. Then, higher-domain server B sent this information and the time duration the user was in the coverage area of higher-domain B to the higher-domain server of the user, A. Finally, higher-domain server A forwarded this data to the user home lower-domain server, A1.

As described above, the home lower-domain server of a user stores the position data of the user all the time, but the information precision differs depending on where the user has been. When the user has been in its home domain, the position data include the user tag ID, the readers' IDs to which the user has connected, and the time duration the user has been in the coverage area of each reader. On the other hand, when the user has been in other (foreign) lower-domains under the same higher-domain, the position data include the user tag ID, the identifiers of those foreign lower-domains, and the time duration the user has been in each of the foreign lower-domains. Furthermore, when the user has been in other (foreign) higher-domains, the position data include only the user tag ID, the identifiers of the foreign higher-domains, and the time duration the user has been in each foreign higher-domain. Surely, the home lower-domain server of the user can ask any other lower-domain server under the same higher-domain or any other higher-domain server for more precise information about the user, e.g., to which reader the user has connected and how long the connection has sustained. Since the traffic between two domains in a hierarchical topology, especially between two higher-domains, is limited, the messages transferred over the backbone network can be significantly reduced.

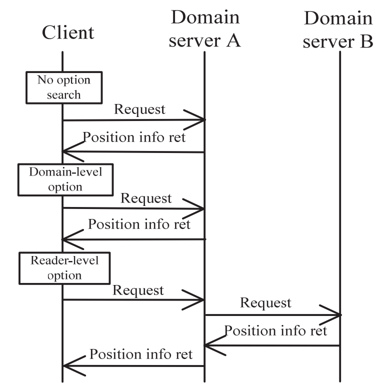

Fig. 11 Position data search sequence in flat topology.

3.5 Position Data Search

The user position data are stored in the database within the users' home domain/lower-domain servers and can be accessed via the Web interface. The position data can also be sent to clients who register to the system by email or by using a push notification service similar to that in the position service using a mobile phone [11]. A client who wants to know the current or historical position of a user submits a request with an option showing the precision degree of the position information the client requires.

A. Position data search in flat topology

The precision degrees of the position information in a flat topology are classified into two levels: *reader* and *domain levels*. The reader level information shows the readers to which the target user was connected. By using the reader-level information and an appropriate estimation approach in [5]–[7], [12], [16], the same precision can be obtained as that in previous approaches. However, the domain-level information shows only the domains wherein the target user has been.

Figure 11 shows some examples of the position search operations for a user who travelled in the order shown in Fig. 7. If a client requests the user position information at the domain-level, the user home-domain server responds to the request immediately since it knows the domain information in which the requested user has been. However, if the client requests the position information at the reader-level, the user home-domain server first asks the domain servers the user has been in for the reader-level information and then sends the information back to the client. The client can also request the position information by default, i.e., without any option, and in this case the domain server only sends back the information it has on hand; that is, the reader-level information when the user was in its home domain and the domain-level information when the user was in foreign domains.

B. Position data search in hierarchical topology

The precision degrees of the position information in a hierarchical topology are classified into three levels: *reader, lower-domain*, and *higher-domain levels*. The reader-level information shows the coverage area of the readers to which the target user was connected. However, the lower-domain-level information shows the lower-domains the target user was in the lower-domains under the same higher-domain and under the foreign higher-domains. Furthermore, the higher-domain-level information shows the higher-domains the target user was in the home higher-domain and in the foreign higher-domains. Similar to the position search in a flat topology, by using the reader-level information and an appropriate estimation approach in [5]–[7], [12], [16], the same precision can be obtained as that in previous approaches. However, the higher-domain and the lower-domain-level information show only the higher-domains and the lower-domains respectively wherein the target user has been.

If a client requests the position information of a user with the higher-domain-level option, the home lower-domain server responds to the request immediately since it knows the higher-domain information where the target user was. However, if the client requests the position information at the lower-domain level, the home lower-domain server first asks each foreign higher-domain server for the information of the lower-domains wherein the target user has been, and then sends this information back to the client. Furthermore, if the client requests the position information with the reader-level option, the home lower-domain server first passes the request to the foreign higher-domains wherein the target user has been, and then each foreign higher-domain server asks each lower-domain server under itself for reader-level information concerning the user. The user home lower-domain server also asks the lower-domain servers under the same higher-domain for the reader-level information concerning the user.

A client can also request the position information of a user without any option, and in this case the home lower-domain server only sends back the information it has on hand. That is, the reader-level information when the user was in its home lower-domain, the lower-domain-level information when the user was in other lower-domains under

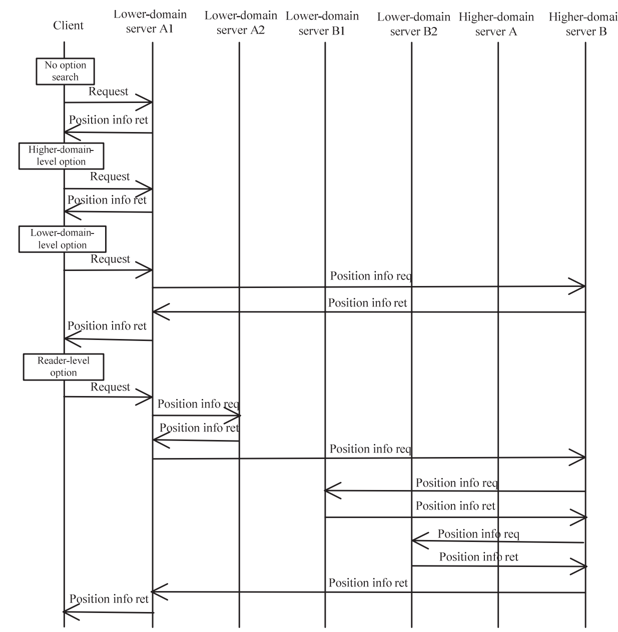

Fig. 12 Position data search sequence in hierarchical topology.

the same higher-domain, and the higher-domain-level information when the user was in foreign higher-domains. Figure 12 shows some examples of the position search operations for a user who has travelled in the order shown in Fig. 9.

4. Implementation of Proposed Approach

The proposed approach has been implemented in an experimental environment that is based on Microsoft Windows and by using ASP.NET, .NET Framework and .NET Compact Framework technologies. Three kinds of wireless devices, e.g., Wi-Fi tags [3], RFID tags [19], and PDAs [9], were used as mobile terminals in the implementation. The format of a tag ID is given in a format similar to “HHH+LLL+serial_no” and is used as a tag identifier where “HHH” and “LLL” indicate the higher- and the lower-domain identifiers, respectively. The Internet Protocol (IP) addresses of the lower- and higher-domain servers are used as the identifiers for the corresponding lower- and higher-domains, respectively.

A Web interface was created in the implementation to provide the user position information so that a client can obtain the position information of a target user via a web browser, as shown in Fig. 13. Note that the signal of a tag can be sensed at the same time by more than one reader/access-points as shown in Fig. 13, e.g., an RFID tag was sensed at around 22:13:40 by the readers deployed at both the south and west points in room E311 (See the records with the underlines and marker “*”). Note also that a client does not need to have information about which kind of wireless device the targeted user uses. In this paper, we only implemented position request functions with no option, but it is easy to extend this to create a position search mech-


User Name : Fumiaki Inoue | Search

User Name	Time	Place	Domain	
Fumiaki Inoue	2009/01/22 22:14:18 – 2009/01/22 22:16:34	E311 South	3rd Cluster, Univ. Tsukuba	*
Fumiaki Inoue	2009/01/22 22:13:36 – 2009/01/22 22:14:20	E311 West	3rd Cluster, Univ. Tsukuba	*
Fumiaki Inoue	2009/01/22 22:10:55 – 2009/01/22 22:13:42	E311 South	3rd Cluster, Univ. Tsukuba	*
Fumiaki Inoue	2009/01/22 20:18:13 – 2009/01/22 20:29:01	E311 North	3rd Cluster, Univ. Tsukuba	**
Fumiaki Inoue	2009/01/22 20:15:52 – 2009/01/22 20:17:38	E311 East	3rd Cluster, Univ. Tsukuba	***
Fumiaki Inoue	2009/01/22 20:15:25 – 2009/01/22 20:15:57	E311 South	3rd Cluster, Univ. Tsukuba	*
Fumiaki Inoue	2009/01/22 20:14:30 – 2009/01/22 20:15:16	E311 East	3rd Cluster, Univ. Tsukuba	*
Fumiaki Inoue	2009/01/22 20:13:46 – 2009/01/22 20:14:32	E311 South	3rd Cluster, Univ. Tsukuba	***

Note : position data are shown in time ordering
* : position data of user with WiFi tag
** : position data of user with RFID tag
*** : position data of user with PDA

Fig. 13 Example of user search results.

anism with information level options.

5. Performance Evaluation

We took into consideration two performance indices to evaluate the proposed approach: the time for obtaining the position data of a requested user and the number of messages transferred over the network. The former is a user-oriented performance index while the latter is a system-oriented one. We also investigated the position estimation precision of the proposed approach with previous approaches.

We assume that a client can request the position information of a target user from anywhere over the Internet to the user home-domain server. We also assume that the round-trip times (RTTs) both for a client request and for a data transmission request between any two domain servers are the same and denoted by R. Table 1 lists the RTTs of the previous approach and the proposed one for flat and hierarchical topologies indicated as *Central*, *Flat* and *Hierarchical*, respectively. We see from Fig. 11 that when a client requests the position information of a user with no or the domain-level option in a flat topology, the response time is the same as that of the previous approach. Furthermore, when a user resides in his/her home domain or home lower-domain, the response time is the same as that of previous approaches. The response time is longer only if the client requests the position information with a more precise option, e.g., reader-level, and when the user resides in foreign domains. We see from Fig. 11 the worst response time in the flat topology is near to $2R$, as shown in Table 1. Similarly, we see from Fig. 12 that the worst response time in the hierarchical topology is near to 3R, as shown in Table 1.

Let's investigate the response time to a client request in more detail. Generally, we have more interest in the precise information about a user when he/she is at home than when he/she is away. For example, when a user is at home, it is common to know in which room or which exact point in a room the user is. However, when the user is far away from his/her home, e.g., he/she is overseas, it is usually enough to only know which country or region the user is residing.

Table 1 Response time to client request.

Central	Flat	Hierarchical
R	No/domain level: R Reader level: [R,2R]	No/higher-domain level: R Lower-domain level: [R,2R] Reader level: [R,3R]

Therefore, a request with no precision option should satisfy most cases in a position-tracking service, yielding the same response time as that in previous approaches. Furthermore, the response time is longer than that of previous approach only if the user home-domain server does not have the precise information about the user since the position data obtained from foreign-domain servers are stored in its database.

In order to compare the number of messages transferred over the network, we performed a simulation experiment. In order to examine the proposed approach equally in flat and hierarchical topologies, we used the same network environment. We considered two scenarios: one has 27 readers and the other 1000 readers. Since the two simulation results showed a similar tendency, we only showed in this paper the latter case to save space. In the hierarchical topology of our simulation experiment, we had 10 higher-domains, each of which is connected to one another by the Internet. We also had 10 lower-domains under a higher-domain, each of which was connected to one another and to the higher-domain server by an Intranet. Furthermore, we had 10 readers equipped in each lower-domain that were connected to the domain server by a local area network (LAN). In the flat topology, on the other hand, we had 100 domains that were partitioned into 10 groups. The domains between groups were connected to one another by the Internet, while the domains within a group were connected to one another by an Intranet. Furthermore, similar to a lower-domain in the hierarchical topology, each domain was equipped with 10 readers that were connected to the domain server by a LAN. In order to compare our proposed approach with the previous central server approach wherein

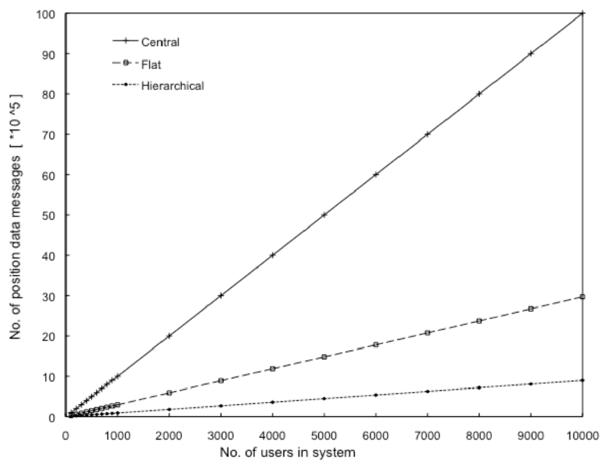

Fig. 14 Position messages passing through Internet.

all the readers are connected to the central server, we considered 1000 readers that were directly connected to the central server by the Internet.

Each simulation was run for 1000 time units and the results were obtained as the average from 100 simulation runs. The number of users in the simulation experiment was increased from 10 to 10,000. Generally, a user spends more time in a limited area around the current position but does not likely move to other places soon. That is, a user stays in the coverage area of the current connecting reader with a larger probability, while moves either to the coverage area of another reader or to another domain with a less probability. We assumed in the simulation experiment that the probabilities that a user remains in the same area covered by the current connecting reader, that the user moves to the area covered by another reader in the same domain, that the user moves to another domain under the same higher domain, and that the user moves to the domain of another higher-domain are 0.4, 0.3, 0.2, and 0.1, respectively.

Figure 14 shows the messages transferred over the Internet obtained in the simulation experiment using the previous approach, denoted by *Central*, and using the flat and the hierarchical topologies in the proposed approach, denoted by *Flat* and *Hierarchical*, respectively. We can see in Fig. 14 that the messages transferred over the Internet both in the flat and the hierarchical topologies are significantly less than that in the previous approach. This is because that a large number of position data transmissions are limited in domains since most users move around near to their current positions. Furthermore, the messages transferred over the Internet in the hierarchical topology can be further reduced compared with these in the flat topology. This is because the position data transmission for a user is limited in his/her current residing higher-domain as long as the user does not leave the higher-domain.

The position estimation precision in the flat and the hierarchical topologies in the proposed approach, denoted respectively by *Flat* and *Hierarchical*, was compared with that in previous approaches as shown in Table 2. The domain server of a user can estimate the user position with the same preciseness as that in previous approaches [5]–[7], [12], [16], if they use the same amount of position information about the user. Therefore, when a user moves around in his/her home domain or home lower-domain, the same position precision can be obtained as that in previous approaches. The same precision can also be obtained if the reader-level option is chosen even when the user is away from his home-domain and residing in a foreign-domain. The precision becomes worse only if the user is away from his home-domain and the domain-level option in the flat topology, or the lower- or the higher-domain-level option in the hierarchical topology is chosen.

Table 2 Comparison of estimation preciseness with previous approach.

	Flat			Hierarchical		
Location	Home	Same		Home	Same	
	Away	Precise degree	Reader-level: same	Away	Precise degree	Reader-level: same
			Domain-level: worse			Lower-domain level: worse
						Higher-domain level: worse

6. Conclusion

We proposed an efficient data management approach in this paper that combines multiple small systems (domains) to form a large-scale system in either a flat or hierarchical topology. The advantages of the proposed approach can be summarized as follows.

- The data formats of various systems are transformed into a unified one so that position data can be processed using the same method.
- The service area for position tracking is expanded to include the entire system covered by all the smaller systems.
- Each domain server stores and processes only the position data of the users residing in its managing domain and therefore the processing loads both in flat and in hierarchical topologies are distributed among the domain servers.
- By providing multiple precision-level options, the traffic for data transmission via the backbone network can be reduced.

We have implemented the proposed approach on an experiment environment using PDAs, RFID tags, and Wi-Fi tags as terminal devices and have shown that the position tracking service can be provided easily using a Web-based user interface. We evaluated the efficiency of the proposed approach through simulation and analysis, and have shown that the proposed approach provides the same estimation precision and response time in most cases and yields limited performance degradation only when users move around away from their home-domains.

References

[1] Y. Arai and S. Naganuma, “Security services based on mobile phone networks: Case in Japan,” Proc. Annual Meeting of Digital Communications, Tallinn, Estonia, 2007.
[2] M. Aso, S. Shimada, T. Yamamoto, N. Sawai, and A. Kariya, “Location information funtions for FOMA terminals — Location positioning function,” NTT DoCoMo Technical J., vol.7, no.4, pp.13–19, 2006.
[3] Ekahau, Inc., “Ekahau positioning engine 4.0,” http://www.ekahau.com/
[4] A. El-Rabbany, Introduction to GPS — The Global Positioning System, 2nd ed., Artech House, Boston, 2006.
[5] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers, “Position estimation for mobile robots in dynamic environments,” Proc. AAAI, Madison, Wisconsin, 1998.
[6] S. Gezici, “A survey on wireless position estimation,” Int. J. Wirel. Pers. Commun., vol.44, no.3, pp.263–282, 2008.
[7] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” Comput., vol.34, no.8, pp.57–66, 2001.
[8] H.G. Hilders, “Safe and secure air cargo containers with integrated RFID visibility system,” A Cargobox International White Paper, CargoBox Holding BV. 2008.
[9] HP iPAQ Pocket PC., http://www.shopping.hp.com/handheld
[10] S. Ito and N. Kawaguchi, “Bayesian based location estimation system using wireless LAN,” Proc. PerCom2005, pp.273–278, 2005.
[11] KDDI Corp., “Anshin-navi,” http://www.au.kddi.com/anshin/index.html
[12] T. Kitasuka, T. Nakanishi, and A. Fukuda, “Design of WiPS WLAN-based indoor positioning system,” Korea Multimedia Society, vol.7, no.4, pp.15–29, 2003.
[13] T. Kurihara, et al., “Sensor network system for municipal crime and disaster prevention using RFID and ubiquitous sensor network technologies,” Matsushita Technical J., vol.52, no.3, pp.72–78, 2006.
[14] A. LaMarca, et al., “Place lab: Device positioning using radio beacons in the wild,” Third International Conference (PERVASIVE2005), Munich, Germany, May 2005, Proc. Series: Lect. Notes Comput. Sci., vol.3468, pp.116–133, Springer, 2005.
[15] R. Malladi and D.P. Agrawal, “Current and future applications of mobile and wireless networks,” Commun. ACM, vol.45, no.10, pp.144–146, 2002.
[16] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, “LANDMARC: Indoor location sensing using active RFID,” Wirel. Netw., vol.10, no.8, pp.701–710, 2004.
[17] NTT DoCoMo Inc., “Map/GPS navigation,” http://www.nttdocomo.co.jp/service/information/index.html
[18] B. Rao and L. Minakakis, “Evolution of mobile location-based services,” Commun. ACM, vol.46, no.12, pp.61–65, 2003.
[19] RF Code Inc., “RF code active RFID spider,” http://www.rfcode.com/
[20] K. Shintani, A. Imai, E. Nishimura, and S. Papadimiriou, “The container shipping network design problem with empty container repositioning,” Transportation Research, Part E, vol.43, no.1, pp.39–59, 2007.
[21] S. Tsuchida, T. Tsushima, K. Aizu, D. Kobayashi, M. Kaji, and Y. Tajika, “NGN-based ubiquitous platform and its application for children’s safety,” Matsushita Technical J., vol.54, no.1, pp.52–56, 2008 (Japanese).
[22] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” ACM Trans. Inf. Syst., vol.10, no.1, pp.91–102, 1992.

Fumiaki Inoue received his B.S. and M.E. degrees in Policy and Planning Science both from University of Tsukuba in 2007 and 2009. He is now a Network Engineer with Softbank Mobile Corp.

Yongbing Zhang received his B.S. in Electrical Engineering from Polytechnic University in 1984, and M.S. and Ph.D. degrees in Computer Science both from the University of Electro-Communications in 1989 and 1992. He was a Research Associate in the Department of Computer Science, at the University of Electro-Communications during 1992–1996. He joined the Institute of Policy and Planning Sciences, University of Tsukuba as an Assistant Professor in 1996 and is now a Professor at the Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include distributed/parallel computer systems, communication networks, and performance evaluation.

Yusheng Ji received her B.E., M.E. and D.E. in electrical engineering from the University of Tokyo in 1984, 1986 and 1989. She joined the National Center for Science Information Systems in 1990. Currently she is an associate professor at the National Institute of Informatics, and the Graduate University for Advanced Studies. Her research interests include network architecture, traffic control and performance analysis for quality of service provisioning in high speed networks. She is a member of IPSJ and IEEE.