SRMS: SIP-based RFID Management System

Kideok Cho†, Sangheon Pack‡, Taekyoung Kwon†, and Yanghee Choi†

†School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: kdcho@mmlab.snu.ac.kr and {tkkwon, yhchoi}@snu.ac.kr

‡School of Electrical Engineering
Korea University, Seoul, Korea
Email: shpack@korea.ac.kr

Abstract— **Radio frequency identification (RFID) is a new technology for object identification/tracking systems. In this paper, we propose a novel RFID location management system: *SIP-based RFID management system (SRMS)*. SRMS employs *session initiation protocol (SIP)*, which is an Internet standard protocol for session management and mobility support. SRMS enhances the existing SIP architecture by introducing a *surrogate user agent (SUA)* and an *SRMS name server (SNS)*. The SUA performs location registration procedures on behalf of RFID tags with limited capabilities, while the SNS provides name resolution services for location registration/tracking of RFID tags. To evaluate the location registration latency in SRMS, we carry out performance analysis based on queueing models. Compared with the existing RFID management systems, SRMS has advantages of service extensibility and reusability.**

I. Introduction

Radio frequency identification (RFID) is a technology that allows an object (e.g., product, animal, or person) to be identified at a short distance using radio frequency. The RFID technology performs object identification without user's intervention, which is one of the representative enabling technologies for ubiquitous computing. Even though the potential for misuse of automatic tracking and privacy concern remain as challenging issues, various types of RFID tags and service platforms on a global scale are now available in commercial markets [1].

Object identification is a well-known application of the RFID technology. So far, bar code systems have been widely used for object identification. However, bar code systems have some limitations against RFID systems. First, a bar code scanner requires a direct line-of-sight (LOS) to extract information from bar codes, and reading multiple tags simultaneously is impossible. Hence, it requires a high cost for manual operations. On the other hand, an RFID reader can read the encoded information even with the non-LOS and multiple items can be scanned at the same time. Second, bar codes can indicate only the basic product information, whereas RFID tags have sufficient storages so that they can store the additional information such as product condition and transportation history.

Since RFID is an identification technology, it should be incorporated into a location management framework to build object identification/tracking systems [2]. So far, a cellular system-based framework [2] and a specialized middleware [3] have been introduced. However, it is expected that diverse network technologies will converge based on the Internet protocol (IP) in the future, i.e., *All IP networks*. Therefore, designing an IP-based RFID management system allows low-cost deployment and easy integration with IP-based services.

In this paper, we propose a novel RFID location management system, called *SIP-based RFID management system (SRMS)*, which is an Internet-friendly solution. Specifically, SRMS uses *session initiation protocol (SIP)* [4] as a basic protocol. SIP is an application layer protocol for session establishment and tear-down, and it is widely used in VoIP services and 3G IP multimedia sub-system (IMS) [5] as a session control protocol. In addition, SIP can support mobility in the Internet [6], [7].

Normally, the hardware capability of an RFID tag is not sufficient to support full IP stacks so that we introduce a surrogate user agent (SUA) and an SRMS name server (SNS) in the SRMS architecture. The SUA filters out multiple readings of a single tag from an RFID reader and performs location registration procedures on behalf of RFID tags with limited capabilities. Also, the SNS is designed for resolution between an RFID identifier (i.e., electronic product code (EPC)) and a SIP universal resource identifier (URI). Compared with the existing systems, SRMS has advantages of reusability and extensibility because it follows the Internet standard architecture.

The remainder of this paper is organized as follows. In Section II, a representative RFID management system proposed by EPCglobal is described. Section III details SRMS and its components. In Section IV, we compare SRMS and EPCglobal Network in terms of reusability, extensibility, and scalability. In Section V, we analyze the location registration latency in SRMS and present numerical results in Section VI. Section VII concludes this paper with future work.

II. EPCglobal Network

The most representative RFID management system is *EPCglobal Network* proposed by EPCglobal [8], which is an organization entrusted by industry to standardize EPC and a real-time/automatic identification system in the supply chain.

1-4244-1326-5/07/$25.00/©2007 IEEE

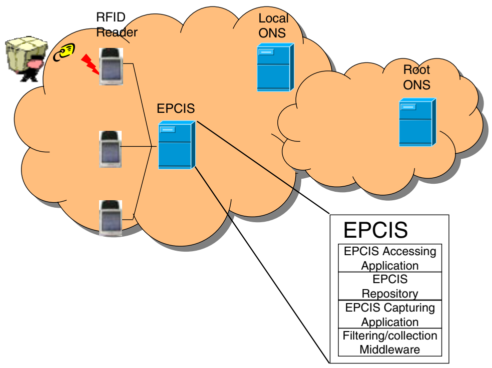

Fig. 1. EPCglobal system architecture.

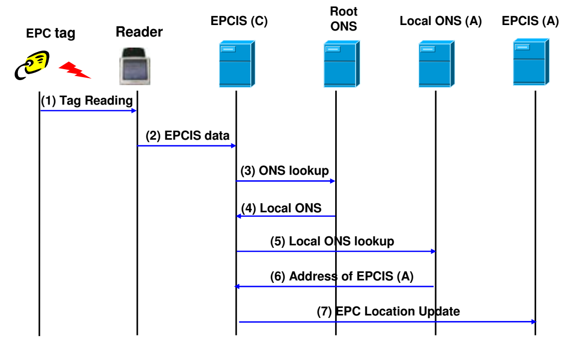

Fig. 2. Location registration in EPCglobal Network.

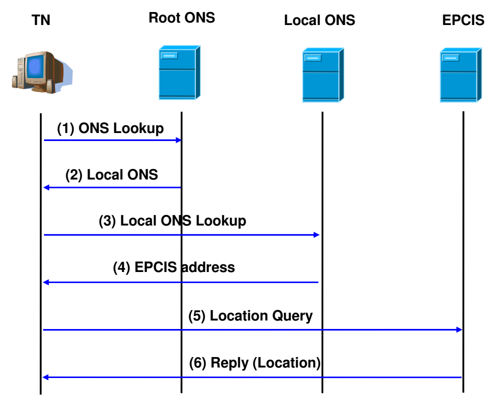

Fig. 3. Location tracking in EPCglobal Network.

Note that EPC is a unique identifier of a physical object stored in a RFID tag.

A. System Architecture

Figure 1 shows the EPCglobal Network architecture. Two main components are *EPC information service (EPCIS)* and *object name service (ONS)*. The EPCIS is a component to exchange data between companies that subscribe to EPCglobal Network and to provide a precise definition of all types of EPCIS data. The EPCIS consists of several components with their own roles and specifies interfaces between them. The components include *RFID reader, filtering/collection middleware*, *EPCIS capturing application*, *EPCIS repository*, and *EPCIS accessing application*. The RFID reader scans EPCs of RFID tags within the range of its antenna and reports the EPCs to the filtering/collection middleware via a reader protocol [9]. The filtering/collection middleware then performs filtering and collecting of received raw-data based on the criteria defined by the EPCIS capturing application. The EPCIS capturing application defines events as well as filtering criteria. The processed EPC information, i.e., events, are stored at the EPCIS repository to be used for future queries from other companies. The EPCIS accessing application carries out overall business decisions such as warehouse management and shipping/receiving, based on the EPC-related data.

On the other hand, the ONS provides a simple lookup service, which takes an EPC as an input and produces a contact address of the EPCIS managing the EPC of interest as an output. The ONS can be hierarchically implemented similar to domain name server (DNS) in the Internet; actually, the ONS can be designed as an application of DNS [10]. In the ONS hierarchy, there are two kinds of ONSs: *root* and *local* ONSs, and the root ONS provides an initial contact point for ONS lookup services [3].

B. Location Registration Procedure

Hereafter, an EPCglobal Network entity in domain X is referred to as *EPCglobal Network entity (X)*. As shown in Figure 2, the following steps are performed for location registration in EPCglobal Network.

Step 1: When a laptop arrives at a warehouse, an RFID reader identifies the EPC of the RFID tag attached to the laptop.

Step 2: The identified EPC is delivered to the EPCIS (C), where the EPC is filtered and temporally stored.

Step 3: The EPCIS (C) then finds the corresponding EPCIS (A) that maintains information about the laptop, by consulting the root ONS.

Step 4: On receiving a lookup request, the root ONS delegates further lookup procedures to the local ONS (A), which resides in the domain (A).

Step 5: The EPCIS (C) then requests the local ONS (A) to find out the address of the EPCIS (A).

Step 6: The local ONS (A) replies with the EPCIS (A)'s address.

Step 7: The EPCIS (C) updates the current location of the laptop by sending a location registration message to the EPCIS (A).

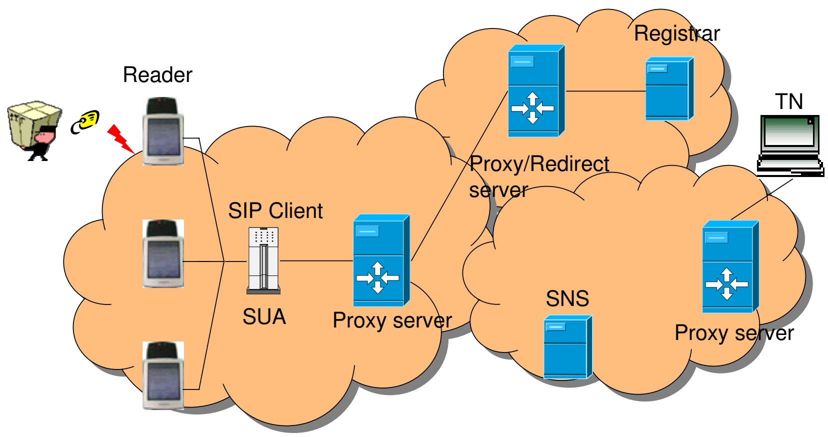

Fig. 4. SRMS system architecture.

C. Location Tracking Procedure

Figure 3 illustrates a location tracking procedure in EPC-global Network. Detailed procedures are as follows.

- **Step 1-4**: To obtain the address of the corresponding EPCIS, a tracking node (TN) consults the root ONS and local ONS, which follows the same procedures as illustrated in the location registration procedure.
- **Step 5**: After finding the corresponding EPCIS, the TN sends a location query to it.
- **Step 6**: On receiving this query, the corresponding EPCIS replies with the current location of the laptop.

III. SRMS: SIP-based RFID Management System

SRMS makes it possible to track an RFID tag's current location at the resolution of the RFID reader level. That is, a TN can learn the identifier of the RFID reader around which the tag resides. To this end, each RFID reader is assumed to have an IP address. Message exchanges for location registration and tracking are based on SIP. However, RFID readers are expected to perform only limited functions such as tag reading and collision handling, and thus a surrogate user agent (SUA) in charge of SIP signaling is introduced. Also, we propose an SRMS name server (SNS), which provides EPC-resolution services.

A. System Architecture

The SRMS system architecture is depicted in Figure 4. The typical SIP architecture consists of SIP servers and user agents. SIP servers are classified into *proxy*, *redirect*, and *registrar* servers depending on their functions. The proxy server relays received SIP messages to other SIP server or user agents, while the redirect server performs redirection of received SIP messages [4]. The registrar maintains location information to support mobility, so that it is employed to manage the RFID tags' current locations in SRMS. Each RFID tag has its own home registrar at its home domain. When the RFID tag moves to a foreign domain, it should perform a location registration procedure to inform its home registrar of the up-to-date current location. Therefore, a TN can consult the registrar to find out an RFID tag's location.

On the other hand, user agents are classified into user agent client (UAC) and user agent server (UAS). The UAC initiates SIP transactions by sending a request message, while the UAS responds with reply messages with suitable status codes [4]. The user agent registers its location at the registrar before establishing a SIP session. The location registration procedure is a core operation regarding RFID management. However, since RFID tags have limited capabilities, it cannot be expected that RFID tags perform SIP operations by themselves and therefore we introduce the SUA and the SNS in the SRMS architecture.

B. Surrogate User Agent (SUA)

The SUA is a SIP user agent that performs SIP signaling on behalf of RFID tags with limited capabilities. The SUA fulfills three functions: *EPC filtering*, *tag-location registration*, and *handling tag-tracking requests*. The EPC filtering performs a similar function to the filtering/collection middleware in EPCglobal Network. Since the RFID reader has only a simple scanning function, there may exist multiple readings for one tag. To resolve this redundancy, when an RFID reader scans an RFID tag, it sends scanned information to the SUA. On receiving the information, the SUA first performs two filtering processes: temporal filtering (i.e., whether it is redundant EPC or not) and product filtering (i.e., whether it is EPC of interest or not). After two filtering processes, the SUA triggers a location registration procedure on behalf of the RFID tag. To do this, the SUA needs a SIP URI for the RFID tag and thus it obtains the SIP URI by referencing the SNS. After obtaining the SIP URI, the SUA updates the current location of the RFID tag by sending a `REGISTER` message to the registrar in the tag's home domain. The current location, i.e., the SUA's domain name, is contained in the `contact` field

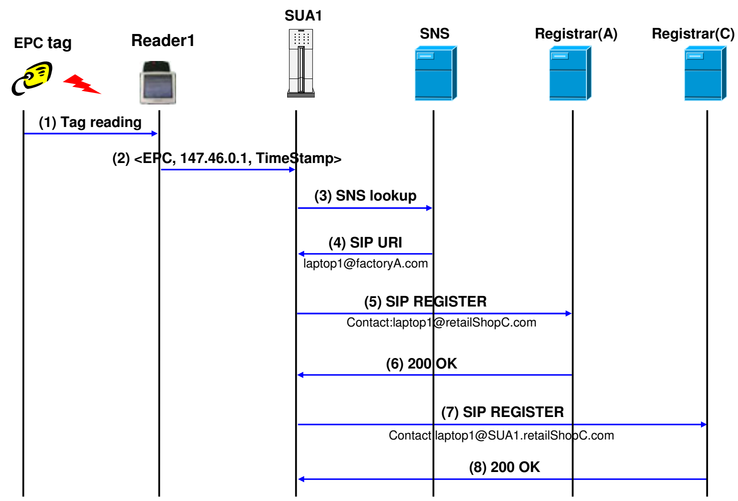

Fig. 5. Location registration procedure in SRMS.

of the REGISTER message.

For location tracking of RFID tags, the SUA also plays a crucial role. If a TN wants to track the current location of an RFID tag, the TN sends an INVITE message to the corresponding SUA. Then the SUA replies with the tag's current location. In other words, if a tracked tag resides under a SUA's administration, the SUA informs the TN of the tag's current location by sending a 200 OK message which contains the reader's IP address in the contact field. On receiving this 200 OK message, the TN learns the tag's current location by parsing the contact field. In addition, if the TN wants to monitor the tag's condition, it can receive data (e.g., video clip) from the SUA by exploiting the established SIP session.

C. SRMS Name Server (SNS)

The SNS is a distributed database which maintains mapping between EPCs and SIP URIs. The SNS provides lookup service similar to DNS and ONS; i.e., the SNS takes an EPC as an input and returns a SIP URI. The SUA or TN sends the concerned tag's EPC to the SNS in order to obtain the SIP URI of the tag for location registration or tracking purposes. Then, the SNS finds out the SIP URI for the received EPC and returns it to the SUA or TN.

The EPC is divided into three parts: general manager number, object class, and serial number [12]. In the case of 96-bit EPC encoding, EPC consists of 8-bit header, 28-bit general manager number, 24-bit object class, and 36-bit serial number. General manager number uniquely identifies a company or an organization, which, in turn, is responsible for maintaining subsequent parts, i.e., object class and serial number. Object class and serial number are used to uniquely identify a product. The same type of products has the same object class but has different serial number. The general manager number of an EPC is used to generate home domain in an SIP URI. Namely, the general manager number tells the manufacturer of a product and thus it enables the SNS to find out the home domain (e.g., factoryA.com) of the corresponding RFID tag. After then, SNS learns the product type (e.g., laptop) by examining the object class. Also, the serial number is used to identify an RFID tag uniquely. As a result, the SNS can generate an SIP URI in the form of *laptop1@factoryA.com*.

Due to performance and practical reasons, the SNS will be deployed in distributed and hierarchial manners. In this case, when requesting lookup services to the SNS, an SUA or a TN first asks a local SNS. If the local SNS can resolve the request, it replies with an SIP URI. However, if the local SNS have no information about tag's home domain, it can't resolve this query; it can't make an SIP URI. Then, the local SNS asks other SNS recursively which is at the higher level in the SNS hierarchy to resolve this query. When another SNS at higher level can resolve this query, it delivers the result to the local SNS and the local SNS relays this result to the SUA or the TN. Also, this resolution result can be cached for possible future queries.

D. Location Registration Procedure

For an illustrative example, it is assumed that the concerned RFID tag's home domain is *factoryA.com* and the tag's SIP URI is *laptop1@factoryA.com*. In addition, SUA1 is an SUA which resides in the foreign domain, *retailShopC.com* and the domain name of SUA1 is *SUA1.retailShopC.com*. The RFID tag moves from *factoryA.com* to *retailShopC.com* and it is identified by RFID reader1 whose IP address is 147.46.0.1 and RFID reader1 is under SUA1's administration. The location registration procedure is illustrated in Figure 5, where an SRMS entity in domain X is referred to as *SRMS entity (X)*.

Step 1: RFID reader1 scans an RFID tag attached to a laptop and learns the laptop's EPC.

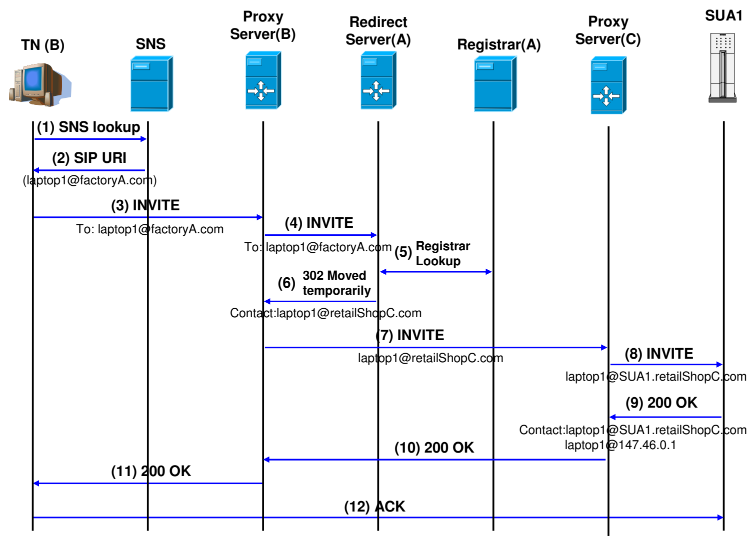

Fig. 6. Location tracking procedure in SRMS.

Step 2: The reader1 forwards the read EPC and related information (e.g., its own IP address and time-stamp for a filtering purpose) to SUA1. Receiving the EPC-related information, SUA1 performs two filtering processes, where SUA1 checks whether there are redundant readings or not and the laptop is an interesting product or not. To reduce overhead in per-tag processing, multiple readings are buffered during a time interval and they are delivered to the SUA as a form of a batch [11].

Step 3: After completing filtering processes, SUA1 consults the SNS to obtain the SIP URI corresponding to the EPC.

Step 4: The SNS finds out the SIP URI and then returns the SIP URI (i.e., *laptop1@factoryA.com*) to SUA1.

Step 5: SUA1 performs a location registration by sending a `REGISTER` message which contains the current location of the tag, *laptop1@retailShopC.com*, in the `contact` field.

Step 6: If the `REGISTER` message is successfully processed, the registrar (A) sends a response message, `200 OK`, back to SUA1.

Step 7: SUA1 also registers the current location of the tag to its local registrar, i.e., registrar (C) in *retailShopC.com*. The aim of this registration is to forward tracking messages which arrives at *retailShopC.com* domain to the tag's current location, i.e., SUA1. Unlike the previous `REGISTER` message destined for the registrar (A), this `REGISTER` message's `contact` field describes more detailed tag's location, *laptop1@SUA1.retailShopC.com*. By this message, the registrar (C) learns precise location of the tag and future SIP messages for the tag can be forwarded to SUA1.

Step 8: The registrar (C) sends a `200 OK` message to SUA1.

As shown in Figure 5, SRMS supports hierarchical location update [7]. Therefore, SRMS can reduce the amount of SIP signalling as similar to Hierarchial Mobile IPv6 [13]. For instance, if an RFID tag moves within the same domain, *retailshopC.com*, a new SUA updates only its local registrar (C), while the home registrar (A) is not updated.

E. Location Tracking Procedure

Figure 6 illustrates the location tracking procedure invoked by a TN in domain B. The followings are the detailed location tracking procedures.

Step 1: A TN that would like to know the current location of a laptop first consults the SNS to finds the laptop's SIP URI.

Step 2: The SNS finds out the corresponding SIP URI and returns it to the TN.

Step 3: With the found SIP URI, *laptop1@factoryA.com*, the TN sends an `INVITE` message to a proxy server (B) in its domain.

Step 4: The proxy server then forwards the `INVITE` message to the redirect server (A) in the *factoryA.com*.

Step 5: Receiving the `INVITE` message, the redirect server (A) queries the registrar (A) to find the current location of the laptop. For lookup process, directory management protocols such as lightweight directory access protocol (LDAP) [14] can be employed.

Step 6: If the correspondent laptop has moved to other domain, say *retailShopC.com*, the redirect server (A) informs the proxy server (B) of the movement. This notification is accomplished by sending a `302 moved` temporarily response message, which contains the laptop's new SIP URI, *laptop1@retailShopC.com* in contact field. This new SIP URI has already been registered through a location registration procedure when the laptop moved to the *retailShopC.com* domain.

Step 7: After receiving the response message, the proxy server (B) sends an INVITE message again to proxy server (C) in the *retailShopC.com* domain.

Step 8: By consulting the registrar (C), the proxy server (C) learns that SUA1 manages the RFID reader which scanned the corresponding RFID tag. Now, the proxy server (C) sends an INVITE message to SUA1 by using the learned SUA1's domain name, *SUA1.retailShopC.com*.

Step 9: SUA1 sends back a 200 OK response message. To track the current location of the laptop at the RFID reader level, the contact field of the 200 OK message includes the IP address of the RFID reader (i.e., 147.46.0.1) as well as the tag's SIP URI (i.e., *laptop1@SUA1.retailShopC.com*).

Step 10-11: Receiving this response, the proxy server (C) relays the 200 OK message to the proxy server (B), which in turn relays the response to the TN.

Step 12: At last, the TN sends back the ACK message to SUA1.

With the received 200 OK message, the TN learns the current location of the laptop by parsing the contact field in the message. In addition, if the TN wants to monitor the laptop's condition, it can receive data (e.g., video clip) from the SUA1 by exploiting the established SIP session.

IV. Discussions

In this section, we will compare SRMS with the EPCglobal proposal, in terms of reusability, extensibility, and scalability.

First, SRMS can reuse the existing network infrastructures, e.g., SIP-based VoIP networks. To manage RFID tags, EPCglobal Network requires a separate network architecture and components. On the other hand, since SRMS uses SIP without any modification, SRMS reuses the existing SIP elements such as SIP proxy server, redirect server, and registrar. Currently, SIP is widely employed for a signaling protocol (e.g., VoIP services, IP multimedia sub-system (IMS) [5]). Accordingly, it is possible to integrate SRMS with the existing infrastructures and hence this reusability allows low deployment cost and quick development.

Second, SRMS is extensible. EPCglobal Network is designed as a middleware solution with a special goal. Hence, it has limitations in service extension, e.g., interworking with web services and mobile Internet. On the other hand, SRMS is based on SIP which uses text-based encoding similar to HTTP, and therefore new services can be easily added at the application-level or user-level. Also, SRMS can be easily extended and integrated with other Internet-based services via SIP.

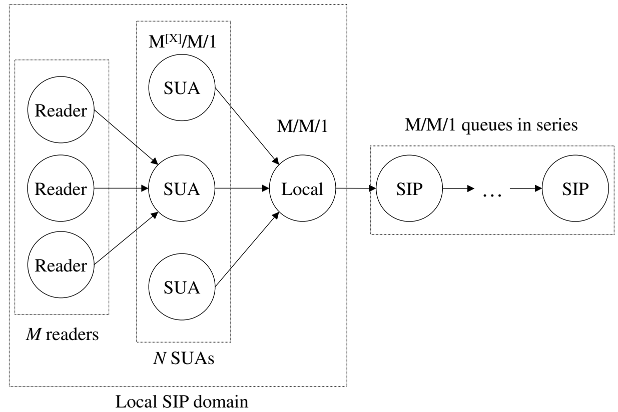

Fig. 7. System model.

Third, SRMS is highly scalable. In SRMS, there is no central server to manage SIP servers and SNS. SIP servers (i.e., proxy and redirect servers) are distributed in multiple domains and each server performs its own functions in its domain. Also, the SNS can be implemented in a distributed manner as the DNS does. Similar to the DNS query resolution, EPC lookup in SRMS can be resolved in a local SNS first, and if not, it will be resolved at the higher level in the SNS hierarchy. On the contrary, ONS operates in a centralized manner. Even though the ONS is divided into two-level hierarchy, i.e., root ONS and local ONS, the root ONS is the initial point of contact for ONS lookup [3]. Therefore, all lookup resolutions for EPC start by sending queries to the root ONS, which makes the root ONS a single point of performance bottleneck.

V. Performance Analysis

To evaluate the performance of SRMS, we model the location registration latency, which is defined from the time instance that a tag is scanned to the time instance that an INVITE message for the tag arrives at the home registrar. Figure 7 shows the system model under consideration. In a SIP domain, M readers are managed by an SUA and N SUAs are managed by a local proxy server. Scanned information of X distinct tags is transmitted to the SUA as a unit. Therefore, the SUA can be modeled by a queue with batch arrival, i.e., $M^{[X]}/M/1$ queue and it is assumed that X follows a geometric distribution and its probability density function (pdf) is $f(x) = p(1-p)^{n-1}$. For location registration, the SUA sends INVITE messages to the local proxy server. Since the departure process of an $M^{[X]}/M/1$ queue can be approximated to a Poisson process, the local SIP server is modeled as an $M/M/1$ queue. On the other hand, intermediate SIP servers and destination registrar are modeled by $M/M/1$ queues in series.

The location registration latency in SRMS is given by

$$L = t + E[SUA] + E[Local] + K \times E[SIP], \qquad (1)$$

where $E[SUA]$ and $E[Local]$ are the processing delays at the SUA and local proxy server, respectively, and t is propagation

TABLE I

PARAMETERS FOR NUMERICAL ANALYSIS.

Parameter	Value
Propagation delay (t)	40 msec
Service rate of the SUA (μ_{SUA})	125 tags/sec
Service rate of the local proxy server (μ_{local})	100 tags/sec
Average service time of a remote SIP server ($E[S]$)	10 msec
Utilization of a remote SIP server (ρ_{SIP})	0.5
The number of remote SIP servers (K)	5
The number of SUAs (N)	4

delay. K is the number of remote SIP severs including intermediate SIP servers and destination registrar. $E[SIP]$ is the processing delay at the remote SIP server.

Let λ_{SUA} and μ_{SUA} be the arrival and service rates of the SUA $M^{[X]}/M/1$ queue. By [15], the processing latency at the SUA is computed as

$$E[SUA] = \frac{E[X] + E[X^2]}{2\mu_{SUA}(1 - \rho_{SUA})E[X]}, \quad (2)$$

where $E[X]$ and $E[X^2]$ are the first and second moments of batch size, respectively. ρ_{SUA} is the utilization of the SUA and it is given by

$$\rho_{SUA} = \frac{\lambda_{SUA}E[X]}{\mu_{SUA}}. \quad (3)$$

Since the local proxy server is modeled as a $M/M/1$ queue, $E[Local]$ is given by

$$E[Local] = \frac{1}{\mu_{Local} - \lambda_{Local}}, \quad (4)$$

where λ_{Local} and μ_{Local} are the arrival and service rates of the local proxy server. A local proxy server covers N SUAs and the inter-departure process of an SUA is approximated to an exponential distribution with rate $\frac{\mu_{SUA}\lambda_{SUA}}{\mu_{SUA}+\lambda_{SUA}-\lambda_{SUA}E[X]}$ (see Appendix I). Therefore, λ_{Local} is given by

$$\lambda_{Local} = N \times \frac{\mu_{SUA}\lambda_{SUA}}{\mu_{SUA} + \lambda_{SUA} - \lambda_{SUA}E[X]}. \quad (5)$$

Let $E[S]$ and ρ_{SIP} be the average service time and utilization of a remote SIP server, respectively. Then, the average processing time at the remote SIP server is expressed as

$$E[SIP] = \frac{E[S]}{1 - \rho_{SIP}}. \quad (6)$$

VI. NUMERICAL RESULTS

Table I shows the parameter values used for numerical analysis [2], [16]. Figure 8 indicates the location registration latency (L) as a function of the SUA's arrival rate (λ_{SUA}). As the arrival rate of the SUA increases, the registration latency increases due to the increased processing latency. Also, when $E[X]$ is large, the SUA should perform more registration procedures, so that the registration latency is also long.

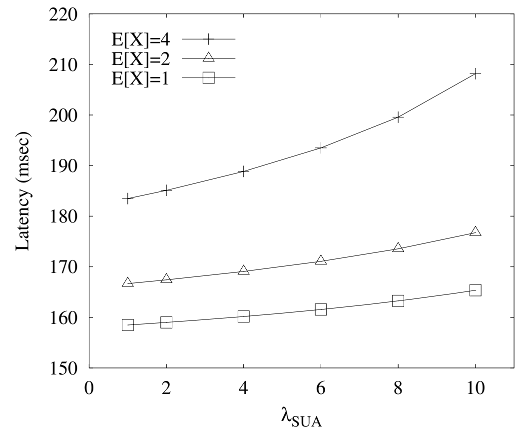

Fig. 8. Location registration latency vs. λ_{SUA}.

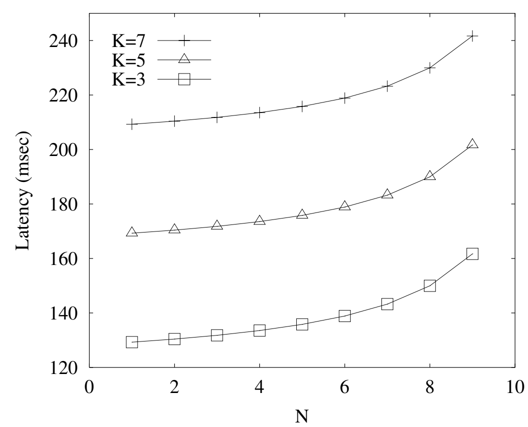

Fig. 9. Location registration latency vs. N.

The effects of N and K are illustrated in Figure 9. Intuitively, as N increases, the registration latency increases because more SIP messages should be processed by a local proxy server. Therefore, it is recommended to deploy multiple local proxy servers in a large domain where many SUAs exist. On the other hand, a large K represents that the home registrar is located far away. Hence, the registration latency increases with the increase of K. This observation indicates that it is possible to reduce the location latency by employing hierarchical registration [7].

VII. CONCLUSION

In this paper, we have proposed a SIP-based RFID management system (SRMS). SRMS is based on a standard Internet protocol, SIP, which provides an efficient mobility solution. In SRMS, the surrogate user agent (SUA) is introduced to handle SIP-related messages and tag filtering. In addition, the SRMS name server (SNS) is used to provide global naming services between EPCs and SIP URIs. Compared to EPCglobal Network which is a representative RFID management system, SRMS has the advantages of reusability, extensibility, and scalability because it is based on a distributed architecture using SIP. We are currently building an SRMS testbed using partysip SIP proxy server [17] and we will conduct comprehensive performance study over the testbed. Also, we will investigate interoperbility and security issues in the large-scale SRMS.

ACKNOWLEDGMENT

This work was supported in part by the BK 21 project of the Ministry of Education and in part by grant No. R01-2004-000-10372-0 from the Basic Research Program of the KOSEF, 2006, Korea. The ICT at Seoul National University provides research facilities for this study.

APPENDIX I
DERIVATION OF INTER-DEPARTURE TIME PROCESS

Let $A(s)$ and $B(s)$ be the Laplace transforms of inter-arrival time and service time at the SUA, respectively, and they are given by

$$A(s) = \frac{\lambda_{SUA}}{s + \lambda_{SUA}} \quad \text{and} \quad B(s) = \frac{\mu_{SUA}}{s + \mu_{SUA}}.$$

Then, the Laplace transform of the inter-departure time, $D(s)$, is derived as

$$\begin{aligned} D(s) &= (1 - \rho_{SUA}) \left(\frac{\lambda_{SUA}}{s + \lambda_{SUA}} \right) \left(\frac{\mu_{SUA}}{s + \mu_{SUA}} \right) \\ &+ \rho_{SUA} \left(\frac{\mu_{SUA}}{s + \mu_{SUA}} \right). \end{aligned} \tag{A.1}$$

Since ρ_{SUA} is equal to $\lambda_{SUA} E[X]/\mu_{SUA}$, $D(s)$ can be rewritten as

$$D(s) = \frac{\lambda_{SUA}(sE[X] + \mu_{SUA})}{(s + \mu_{SUA})(s + \lambda_{SUA})}. \tag{A.2}$$

Consequently, the average inter-departure time is computed as

$$\overline{T} = (-1) \left. \frac{dD(s)}{ds} \right|_{s=0} = \frac{\mu_{SUA} + \lambda_{SUA} - \lambda_{SUA} E[X]}{\mu_{SUA} \lambda_{SUA}} \tag{A.3}$$

and the average inter-departure rate is $1/\overline{T}$.

REFERENCES

[1] R. Want, "Enabling Ubiquitous Sensing with RFID," *Computer*, IEEE, vol. 37, pp. 84-86, April 2004.

[2] P. De, K. Basu, and S. Das, "An Ubiquitous Architectural Framework and Protocol for Object Tracking using RFID Tags," in *Proc. ACM Mobiquitous 2004*, August 2004 .

[3] EPCglobal, "The EPCglobal Architecture Framework," EPCglobal Standard Specification, July 2005.

[4] J. Rosenberg et al. "SIP: Session initiation protocol," RFC 3261, IETF, June 2002.

[5] G. Camarillo, M. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS) Merging the Internet and the Cellular Worlds*, John Wiley & Sons, August 2004.

[6] N. Banerjee, S. Das, S. Dawkins, and J. Pathak, "Mobility Support in Wireless Intenet," *IEEE Wireless Communications*, October 2003.

[7] E. Wedlund, H. Schulzrinne, "Mobility support using SIP," in *Proc. ACM WoWMoM 1999*, August 1999.

[8] EPCglobal Website, http://www.epcglobalinc.org.

[9] EPCglobal, "Reader Protocol 1.0," EPCglobal Working Draft, March 2005.

[10] EPCglobal, "EPCglobal Object Name Service (ONS) 1.0," Ratified EPCglobal Standards, October 2005.

[11] EPCglobal, "The Application Level Events (ALE) Specification, Version 1.0," Ratified EPCglobal Standards, September 2005.

[12] EPCglobal, "EPC Generation 1 Tag Data Standards Version 1.1 rev 1.27," EPCglobal Standard Specification, May 2005.

[13] H. Soliman, et al, "Hierarchical Mobile IPv6 mobility management (HMIPv6)," RFC4140, IETF, August 2005.

[14] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)," IETF RFC 2251, December 1997.

[15] D. Gross and C. Harris, *Fundamentals of Queuing Theory*, John Willey & Sons, 1985.

[16] Proceedings of the fifty-third Internet Engineering Task Force, http://www3.ietf.org/proceedings/02mar/slides/ippm-4.pdf.

[17] Partysip SIP Proxy Server, http://www.partysip.org.