

# A Dynamic Service Lookup and Discovery Scheme using a Self-Organizing Overlay Network for Indoor Location-Based Service

Seol Young Jeong, Hyeong Gon Jo, Soon Ju Kang\*  
 School of Electronics Engineering, College of IT Engineering  
 Kyungpook National University  
 Daegu, Republic of Korea  
 e-mail: { snowflower, tsana, sjkang\* } @ee.knu.ac.kr

**Abstract**— Indoor location-based service (LBS) has various challenges, including that of numerous service lookups being requested concurrently and frequently from several locations, and that the network infrastructure needs to support high scalability, such as inserting or deleting network nodes anytime and anywhere. In general, indoor LBS resources are generally located in close proximity to the requested point. However, a traditional centralized LBS system needs to maintain a geographical map of the entire building or complex in its central server, which can cause low scalability and traffic congestion. This paper presents a self-organizing and fully distributed indoor LBS platform through regional cooperation among devices, and a service lookup algorithm that searches for the shortest physical path to the service resource. An evaluation of the performance of the proposed platform has been compared to the traditional centralized method regarding the service turnaround time according to the number of concurrent lookup increases.

**Keywords**- overlay network; indoor LBS; self-organizing; distributed system; resource lookup

## I. INTRODUCTION

Indoor location based services (LBS) are gradually becoming prevalent due to the increasing numbers of smartphone users. In particular, various intelligent building systems provide indoor LBSs to users, managed by a central server. However, this centralized architecture has several disadvantages: it requires global knowledge to construct and maintain a complete geographical indoor map, the central server is a single point of failure due to its centralized search mechanism, and traffic congestion can occur when numerous lookups are concurrently requested from several different points.

In order to solve the challenges inherent to the centralized server architecture, a location-based self-organizing fully distributed network infrastructure needs to be designed, similar to a natural spontaneous environment with a lot of movement, such as an ant or bee colony. This self-organizing and fully distributed indoor LBS system would not need to maintain or employ a centralized map and

would be dynamically adaptive to changes in the indoor infrastructure. Furthermore, personal privacy would be protected, without the need to forward the user's data to the central server; the requested data would only be transferred within the local range surrounding the point of the lookup activity.

This paper proposes an overlay network infrastructure – called the self-organizing service platform (SOSp) and the related service lookup algorithm based on the SOSp. The self-organizing service router (SOS-Router), which works as a unit member of the proposed overlay network [1] on top of the legacy physical network, is installed in a unit space (i.e. a room or a corridor). The SOS-Router has all of the service resource information, such as printers, medical devices, etc. in its unit space, as shown in Fig. 1.

Fig. 1 shows an example of an indoor environment and service scenario. There are numerous resource devices either mobile or stationary, with the mobile devices like smartphones being able to request the following service examples:

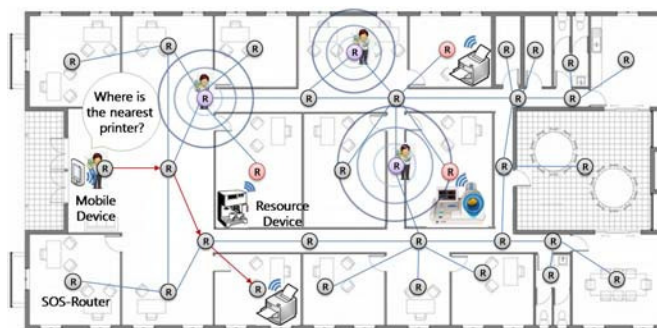


Figure 1. An example of an indoor LBS environment and the proposed service scenario.

- A mobile device user can print to the nearest physical printer (resource device) from his current position without presetting (zero-configuration).
- After the user uses a local sphygmomanometer, he can look-up and check his own medical data, such as

blood pressure, anywhere and anytime, regardless of his current location without current location without the need to configure any settings.

In order to provide these services, the SOS-Router works as a stationary node and so represents the unit space location. Therefore the SOS-Router consolidates the unit's service resources, as shown in Fig. 1. Furthermore, the SOS-Router manages the connection between a service resource, i.e. a printer, and a mobile device, such as a smartphone. The primary purpose of this research is to provide LBSs between the resource devices and mobile devices on this overlay network.

In particular, numerous mobile devices can request LBSs from several locations, as shown in Fig. 1; each SOS-Router represents the position used to request the service works located within its range. In this manner the overall network architecture is fully decentralized and localized. In addition, an expansion of the service area or an increase in the number of mobile or resource devices does not cause a growth in the complexity, unlike the traditional centralized server-based architecture.

This paper discusses the function of priority search resources for the nearest neighbor from the current position of a mobile device requesting an LBS. The proposed architecture provides fully distributed LBSs, in that it does not have a concentration of unnecessary traffic, and is less sensitive to service area expansion and an increased number of devices (both mobile devices and resource devices). Furthermore, this architecture improves the scalability of the infrastructure, because there is no need to maintain a map of the entire building or complex in the central server. Most importantly, there is no single point of failure due to local traffic overflows or system faults.

The remainder of the paper is structured as follows: Section II presents the detailed design and operation of the proposed indoor LBS lookup system and algorithm. Section III introduces the implementation of various devices in the proposed platform, and demonstrates the performance through simulation. Related research is discussed in Section IV and our conclusions are drawn in Section V.

## II. THE SYSTEM DESIGN AND OPERATION PRINCIPLES

### A. Configuring the SOSp Network Environment and the Software Architecture of the SOS-Router

The proposed SOSp network architecture is comprised of SOS-Routers that represent the unit space, resource devices used to provide physical services, and mobile devices with which the users request necessary services. Fig. 2 shows the proposed software architecture of the SOS-Router and the composition of the SOSp.

Resource devices, such as office equipment, home appliances, and health equipment, have the capability of providing LBS through wired/wireless communication with their SOS-Router, installed with communication modules in the unit space. A user can easily request any indoor LBS service from the physical resources (such as printing, health checking, and hot coffee service) with a mobile device (such as a watch, smartphone, or smart pad) using its embedded

wireless communication function (e.g. WiFi, Bluetooth, etc.). The mobile devices and resource devices always communicate with the SOS-Router in the unit space using the LIDx&AMD [2] protocols, which provides real-time localization and the ability to transfer asynchronous messages amongst numerous mobile nodes, such as mobile devices and service resource devices.

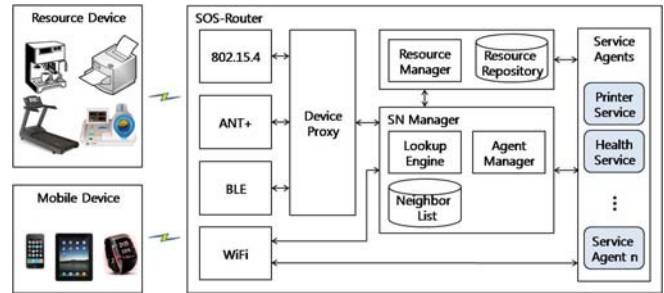


Figure 2. The SOSp composition and SOS-Router software architecture.

The internal software architecture of the SOS-Router, which is the service agent and router for the distributed environment, is comprised of four components: the Device Proxy, the Resource Manager, the Stationary Node (SN) Manager, and the Service Agents. The Device Proxy analyzes the communication data through the wireless communication module, and forwards the analyzed data to the requested Service Agent. The Resource Manager manages and stores the services and resources of the Service Agent running in the SOS-Router to a Resource Repository. The SN Manager, working as the Agent Manager, controls the life cycle of all of the Service Agents capable of running in the SOS-Router. In addition, the SN Manager stores the list of the neighboring SOS-Routers, and provides lookup results based the proposed Lookup Engine, whose details will be discussed in the following subsection. Finally, the Service Agents, which are related to the service resources and service applications for the SOS-Router, manage the requested service data and communicate with resource devices through the wireless communication module in the unit space.

### B. The Operation Principles of the Lookup Engine

Fig. 3 illustrates the initialization process for the initialization in a SOS-Router and the connection to a Service Agent from a mobile device. Initially, the SN Manager configures and initializes the default information of a SOS-Router, and then commences the core services, which are registered in the Resource Repository of the Resource Manager. The mobile devices, which enable the bidirectional communication with the SOS-Router through WiFi or different RF protocols, such as 802.15.4, ANT+, or Bluetooth, with the Device Proxy, have the capability of transferring and receiving data, and requesting services through the connection with the SN Manager. The detailed procedure of lookup by SN Manager shows in Fig. 4. The SN Manager searches for an appropriate Service Agent and forwards a connectable Service Proxy to the mobile device

or the Device Proxy, making the requested services practicable.

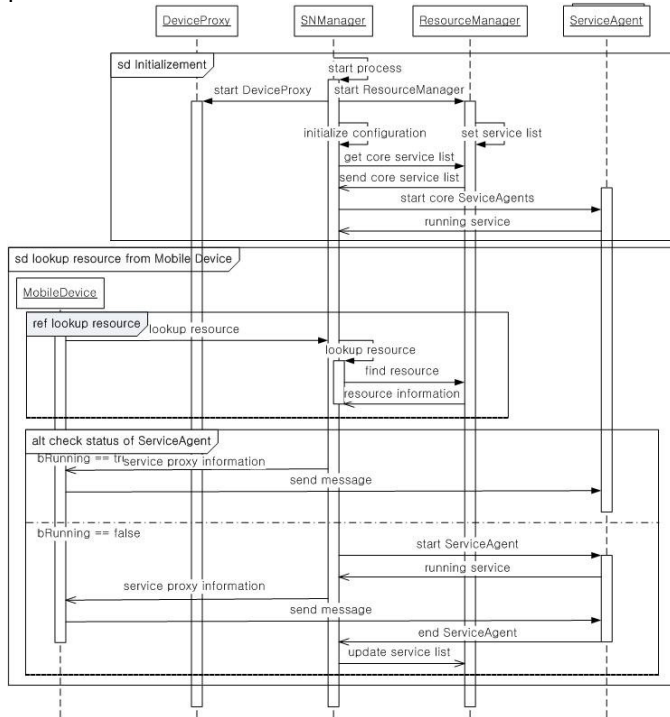


Figure 3. The SOS-Router initialization and connection process for a Service Agent.

Fig. 4 shows the pseudo code of the Lookup Engine in the SN Manager for used to discover the available resources of the nearest physical neighbor to the current position of the mobile device.

**algorithm: Lookup Engine with NSPQ**

NSPQ[] : Neighbor Shortest Path Queue  
tRNQ[] : temporary Request Neighbor Queue  
tnn : temporary neighbor node

```

function lookup(name)
  for i ← 0 to NSPQ[m] // direct lookup
    resource := find(name) in NSPQ[i]
  if resource is none // extend lookup
    for i ← 0 to tRNQ[n-m]
      tnn := get_neighbor_list(tRNQ[i])
      for j ← 0 to NSPQ[m]
        if (tnn ∈ map) && (sum_distance < map_distance)
          map_distance := sum_distance
          exist_nb := true
          break
      if exist_nb is false
        add tnn to map
        resource := find(name) in tnn
        delete i from tRNQ
  return resource
  
```

Figure 4. The indoor LBS lookup pseudo code.

The lookup sequence consists of two parts. First, all of the SOS-Routers have a Neighbor Shortest Path Queue (NSPQ), which is a kind of extendable binary tree. The NSPQ has the order of the neighbor search based on the

shortest path according to the physical path distance from the current position of the mobile node to the location of the next available neighbor SOS-Router. Therefore, the Lookup Engine uses the NSPQ to select the next neighbor when searching for the requested resource. If an available resource is selected by the Lookup Engine, the search result guarantees that the selected resource has the physically shortest distance from the current position of the mobile device. Using this scheme, we can obtain intelligent LBS services, such as “print this document to the nearest printer”.

The NSPQ is extended on every lookup attempt. This means that if there is no instance of a requested resource using the current NSPQ, the NSPQ can be easily extended to include the next neighboring SOS-Router into the tail node of the current NSPQ. This process repeats until the total distance to the neighbor SOS-Router search meets the maximum constraint conditions. Fig. 5 shows the procedure in which the NSPQ is created and extended by request to the neighboring SOS-Router from SOS-Router A. The SOS-Router extends its request in the order of distance through the NSPQ using the neighbor list to the neighboring SOS-Routers, when searching for the requested resources.

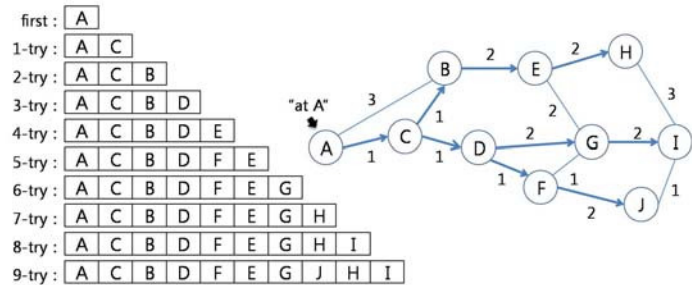


Figure 5. The procedure in which the NSPQ is created and extended.

III. IMPLEMENTATION AND EVALUATION

A. The SOS-Router and Mobile Device Hardware/Software Implementation

SOS-Routers are installed in every unit space of a building, such as a room or a corridor with a wired communication protocol like Ethernet and various wireless communication protocols, such as WiFi, 802.15.4, Bluetooth, ANT+, etc.

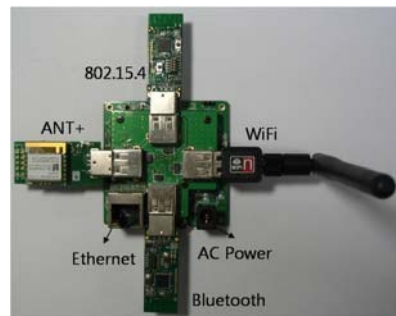


Figure 6. The SOS-Router hardware with its various wireless communication modules.



The SOS-Router hardware, as shown in Fig. 6, is designed to accept various wireless communication modules using USB ports. The SOS-Router utilizes a wired 10/100 Ethernet communication port for the constituent infrastructure of the indoor LBS system, and four USB 2.0 ports for the wireless communication modules.

In order to provide the cooperation needed between the neighbor SOS-Routers and mobile devices using the wired/wireless networks in the self-organizing and fully distributed environment, the internal software and smartphone application components were implemented using Ice [3], a distributed platform middleware. Ice facilitates the development of heterogeneous distributed application by supporting various languages, such as C++, Java, C#, Python, Object-C, Ruby, and PHP over the Windows, Linux, iOS, and Android platforms [3].

There are various types of cell-based SOS-Routers and mobile devices using wireless communication in the infrastructure. The resource devices, such as office equipment, home appliances, and health equipment in the unit space can provide real-time localization using the LIDx&AMD [2] wireless communication protocol developed by our research team. The mobile devices use diverse types of developed dongles for smartphone besides watch, smartphone, or smart pad owned by users. The mobile devices provide the adaptive service using an appropriate profile for the attributes of the devices through the connection with the resource devices. In particular, ID matching with the mobile device owned by a user is important for resources requiring privacy, such as medical and health equipment; the users enable a request to a service using their individual mobile devices.

Fig. 7 shows the communication modules built into the resource devices and various types of mobile devices owned by users.

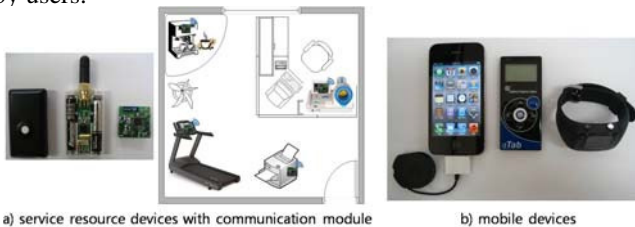


Figure 7. a) Service resource devices and various types of communication modules and b) the mobile devices owned by users.

### B. Tests and Evaluation

In order to evaluate the proposed Lookup Engine with the NSPQ, a simulation environment was configured using 100 SOS-Router processes and numerous requested service packets by mobile devices. Each SOS-Router process used the same application for running the software in the SOS-Router built in the real domain, and included 10 separate service resources. The distances between the SOS-Routers had different values. The simulation program sent the lookup service packets to random SOS-Router processes; the SOS-Routers then determined the requested services using the

proposed lookup algorithm based on the NSPQ, and returned the results.

Fig. 8 shows the average turnaround time needed by the lookup steps at one SOS-Router. The SOS-Router needed time to build the NSPQ in the initial stage, however, the average turnaround time displayed a relatively constant response time in spite of distance differences.

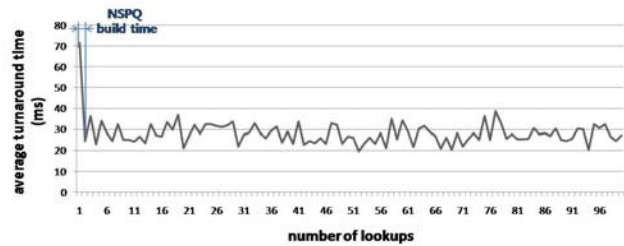


Figure 8. The average turnaround time by lookup steps at one SOS-Router.

Fig. 9 shows the average turnaround time by distance value at one SOS-Router; the response time is directly proportional to the distance to the neighboring SOS-Router. As can be seen in this result, the proposed Lookup Engine employing the NSPQ is highly efficient in distributed environments, due to the fact that the requested services are generally located in close proximity.

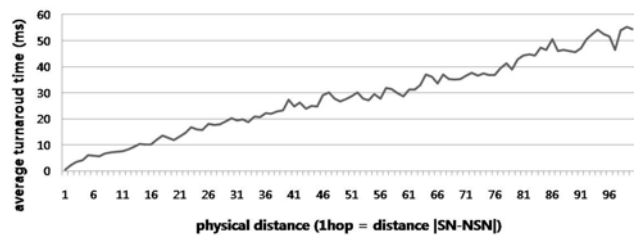


Figure 9. The average turnaround time by distance at one SOS-Router.

Finally, Fig. 10 shows a graph that compares the average turnaround time of the lookup response between the proposed method and the traditional centralized method, when numerous mobile devices concurrently request the service lookup from several locations. We randomly created the service lookups activities from 10 to 10000 in all of the 100 SOS-Routers.

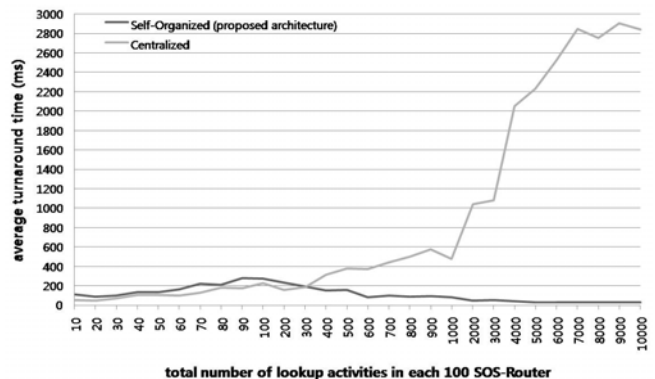


Figure 10. The comparison of the average turnaround time of concurrent searches in all of the 100 SOS-Router.

As shown in Fig. 10, in the case when the total lookup number is under 300, the centralized method is faster than the proposed architecture. However, the performance of the proposed architecture shows comparatively stable results from 30msec to 40msec regardless of the increasing in the frequency of the lookup service requests; the centralized method results explode quickly under the same circumstances.

#### IV. RELATED RESEARCH

Indoor LBSs [4], [5] need to increase their accuracy and precise localization of users and determine an efficient infrastructure topology in an intelligent building. The key technology will discover resources in a dynamic environment that includes numerous resources and users with privacy protection. TraX [6] is an LBS middleware that particularly focuses on position management, advanced functions for interrelating the position data of several targets, and privacy protection. Typical first-generation LBSs are reactive, self-referencing, single-target, and focus on outdoor applications. In contrast, the next generation LBSs (P2P services) will be cross-referencing, multi-target, and self-organizing, which means that position data is exchanged directly between users without the need of intermediary actors.

A P2P overlay network [1] is adequate in an LBS system in that the mobile devices have the capability to ubiquitously access resources offered by resource devices. The first prominent and popular P2P file sharing system, Napster [7], is an example of the centralized model, in which all of the clients send lookup packets to the indexing server in order to locate resources. On the other hand, Gnutella [8], which uses a decentralized searching algorithm, is saturated over portions of the overlay by neighbors forwarding their queries up to a certain radius. This leads to a linear growth of the load on the node as the number of queries increase, thus Gnutella does not scale well. To handle the scaling problem, distributed hash table (DHT) methods have been proposed, such as the Chord [9] system. These existing systems do not offer an appropriate solution due to the fact that their location lookup results do not reflect the geographical shortest path by hop counting based on a physical network router.

In addition, some decentralized model studies include the self-organized and emerging overlay network and have been widely utilized for accomplishing efficient resource discovery in various fields. The self-organizing traffic light system [10] possesses global synchronization adaptively achieved through local interactions between cars and traffic lights, generate flexible green waves on demand unlike centralized legacy traffic lights systems. The Self-Organizing Software System (SOSS) [11] proposed a specific infrastructure for providing arbitrary ambient services at medium-sized locations, like shopping malls, hospitals, and construction sites. This infrastructure together with a supporting software platform that enables service provisioning is called the Ad Hoc Service Grid (ASG), which is based on a distributed algorithm that elects cluster

heads and partitions the network into small clusters. This infrastructure provides self-organized services in a decentralized and self-organized fashion by cluster federation. However, additional overheads are incurred because all of the path nodes need to send unnecessary messages to each cluster head.

#### V. CONCLUSION

This paper presented a self-organizing and fully distributed indoor LBS platform and NSPQ based service lookup algorithm. The cell-based SOS-Router determines the necessary services and resources using only the information regarding its neighbors through cooperation amongst the neighboring SOS-Routers. We also determined that the proposed Lookup Engine with NSPQ is highly efficient over time or as the number of lookups increases.

The proposed SOSp with the SOS-Router platform enhances the scalability, decentralization, fairness, and robustness whenever numerous mobile devices concurrently request service lookups from the several locations. In addition, there is no need to maintain a map of the entire indoor location unlike needed by traditional centralized server methods. Furthermore, the user does not need onerous configuration or setting procedures for the LBS, and personal privacy is also protected because the user data is never gathered into a central server. Future research will focus on extending indoor LBS applications, such as indoor navigation, tracking, mobile asset management, etc.

#### ACKNOWLEDGMENT

"This work was supported by the IT R&D program of MKE/KEIT. [10041145, Self-Organized Software-platform(SOS) for welfare devices]"

#### REFERENCES

- [1] E-K Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," IEEE Communications Survey and Tutorial, March 2004.
- [2] D.K. Lee, T.H. Kim, S.Y. Jeong, and S.J. Kang, "A three-tier middleware architecture supporting bidirectional location tracking of numerous mobile nodes under legacy WSN environment," Journal of Systems Architecture, vol. 57, no. 8, pp. 735-748, Sep. 2011.
- [3] M. Henning and Mark Spruiell, Distributed Programming with Ice. ZeroC, 2003.
- [4] P. Bellavista, A. Kupper and S. Helel, "Location-based services: Back to the future," Pervasive Comput., vol. 7, pp. 85, 2008.
- [5] Gressmann, B., Klimek, H. and Turau, V., "Towards Ubiquitous Indoor Location Based Services and Indoor Navigation," 7th Workshop on Positioning Navigation and Communication (WPNC), pp. 107-112, March 2010.
- [6] A. Kupper, G. Treu and C. Linnhoff-Popien, "TraX: a Device-Centric Middleware Framework for Location-Based Services," IEEE Comm. Magazine, vol. 44, no. 9, pp. 114-120, 2006.
- [7] Napster, <http://www.napster.com>, 1999.
- [8] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network," in Proc. IEEE First Int'l Conf. Peer-to-peer Computing (P2P), 2001.
- [9] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in Proc. SIGCOMM, pp. 149-160, 2001.

- [10] Seung-Bae Cools, Gershenson C., and DHooghe B., "Self-Organizing Traffic Lights: A Realistic Simulation," *Advances in Applied Selforganizing Systems*, Springer London, 2008, pp. 41-50
- [11] Klaus Herrmann, Gero Muhl and Michael Jaeger, "A Self-Organizing Lookup Service for Dynamic Ambient Services," in *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS)*, 2005.