SDRA: A Service Discovery Relocation Algorithm for Multiple Mobile Directories in Wireless Sensor Networks

Thanatporn Kangwanpornsiri, Steven Gordon

School of Information, Computer, and Communication Technology,

Sirindhorn International Institute of Technology, Thammasat University Thailand.

Correspondence:
Thanatporn Kangwanpornsiri
School of Information, Computer,
and Communication Technology,
Sirindhorn International Institute
of Technology, Thammasat University
Thailand
Email: thanatporn.kp@gmail.com,
steve@siit.tu.ac.th

Abstract

Service discovery in wireless sensor networks allows sensors to dynamically locate software/hardware resources in a network without human interaction. For large WSNs, directories are needed to manage service offers and allow clients to locate service providers with minimal search time. However dependence on centralised directories reduces network survivability and introduces performance bottlenecks. This paper proposes using multiple directories in a WSN, and introduces an algorithm to relocate directories to new nodes when needed. The algorithm takes into account connectivity and capabilities of other nodes to select the most appropriate location of a directory. Compared to an existing protocol for WSN service discovery, analysis results show lower communication overhead and response time, while maintaining the successful request rate.

Keywords: centralized architecture, mobile directories, SLP, service discovery, wireless sensor networks

1. Introduction

As the number and capability of wireless devices increases, there will be a variety of different services available in wireless sensor networks (WSNs). The services may be physical sensors (e.g. light, temperature and motion), actuators (e.g. robotic arms, switches, conveyor belts), or software (e.g. data processing and conversions). Service discovery is a mechanism which allows the nodes to automatically locate other nodes that offer software and hardware services. With service discovery in a WSN, nodes do not need to be pre-configured to access particular servers; they can self-configure to find the best service on-demand. This is valuable in WSNs as such networks can be highly dynamic (due to node failure and mobility).

Many protocols and algorithms have been developed for service discovery in wired and wireless networks [1]. The resource constrained environment of WSNs has led to proposals for directory-based service discovery architectures [2]. Nodes with services to offer advertise to a central directory; nodes in search of a service query the directory. (The alternative of not using directories can lead to significant communication overhead due to excessive broadcasting [3, 4]). The challenges in using directory-based service discovery in WSNs include:

- How to handle the failure or overload of directory nodes?
- How to select which node(s) to act as a directory?
- Can directory services be transferred between nodes, e.g. in the case of directory node leaving?
- What lightweight protocol mechanisms should be used so as to minimize overhead in WSNs?

In this paper we focus on the first three problems (the fourth problem is somewhat independent and has been addressed with protocols such as SSLP [5]). Our focus is on using multiple directories to reduce overhead on any single node and mobile directories in order to cope with directory nodes leaving the network. Refs. [1, 6-9] have considered using multiple directories in a WSN. However, they have not shown the tradeoffs in number of directories versus response time and overhead. In [5] and [10], they have explained the methodology of exploiting a single mobile directory. Refs. [1, 7, 10, 11] and our previous work [12] show how to select a directory in the network and study the impact of increasing number of directories. In this paper, we use multiple, mobile directories. The contribution of this paper is the Service Discovery Relocation Algorithm (SDRA) which includes three important features:

- An algorithm for selecting directory nodes based on the capabilities and connectivity of a node.
- The ability to distribute a search to multiple directories in an efficient manner.
- A method for relocating directories from one node to another when either there is a better node than the old one or a directory node leaves.

We evaluate SDRA using simulations and compare it against an approach used in [7]. SDRA can reduce communication overhead, while maintaining high success rate and low response time for requests.

The rest of this paper is organized as follows: In Section 2, we describe the background of service discovery and WSN. Section 3 reviews the related works. In Section 4, the system design which is composed of the proposed algorithms is presented. For analysis, simulations are used to compare with [7]. Section 5 presents the simulation setup, while section 6 gives the results. Section 7 concludes the paper.

2. Service Discovery in WSNs

In service discovery, a *server* offers (hardware or software) *services* for other applications to use, while a client searches for desired servers. Servers may inform other entities of its available service by sending *offer* messages. Clients seek servers by sending *request* messages and receiving *response* messages containing information about the location and features of servers. Optionally, a third entity, a *directory* may act as an intermediary: servers *register* services with a directory; client send requests to a directory as shown in Fig. 1. In this paper, a node that runs a service discovery client is referred to as a *client node*, which is similar for a server and directory.

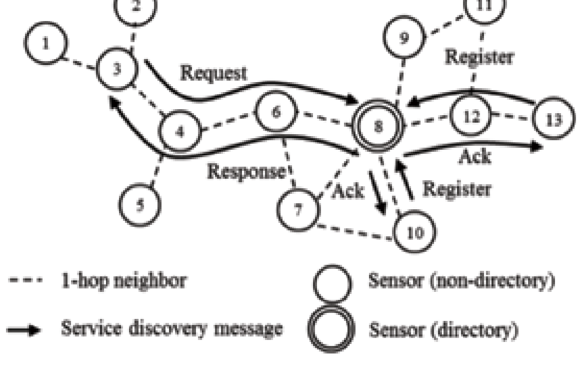

Fig. 1. Directory-based service discovery.

If a directory is not used in service discovery, then either clients broadcast (or multicast) service requests (reactive discovery), or servers broadcast service offers (proactive discovery), or some hybrid combination. Broadcasting messages in WSNs should be minimized so as to reduce communication overhead. Consequently, a directory-less architecture to support service discovery is mainly suited for small networks (up to 10's of nodes) or with infrequent request rates [13, 14]. As the network size grows, the overhead of broadcasting can become excessive. Introducing a directory into the network allows for unicast requests/offers, but at the expense of reliance on a central node.

Service discovery protocols play an important role in WSNs as they have functionality which will enable the device (sensor) to discover a service or offer a service, for instance Jini, SLP, UPnP, Bluetooth, and Salutation. These service discovery protocols were examined by several researchers [1, 15, 16] to compare their properties (security, network topology, process etc.). In our research, we consider a directory-based architecture and use reactive discovery in which any protocol can be supported by this mode; therefore we are not concerned about the specific protocol. In a case of communication overhead we assume the packet size/type by using SSLP protocol [5].

3. Related Work

Several previous works have been proposed to improve service discovery performance in MANETs and WSNs. Most of them have applied the concept of multiple directories to select the appropriate directories, for example, [7] used capability of nodes, [6, 11, 12] manipulated the connectivity of nodes, and [6] took the node direction and mobility pattern into account. These works not only increased performance in a case of reduced overhead, but also saved bandwidth and reduced response time. However, the resources in WSNs and MANETs are limited; therefore, a large number of directories should be considered.

Ref. [5] aimed to find and use services in the close proximity of the user. The proposed algorithm presents proxy directories, which act as directory servers in a 6LowPAN network. Although the scenario differs from ours (we consider only the internal WSN, they focused on service discovery between nodes on an external network and internal WSN). By using multiple proxies inside the WSN, they showed how the impact of a node acting as a proxy directory server failing or leaving the network is reduced. The major drawback of this approach is complexity and the large overhead incurred by translating messages at the gateway between the external and internal network [8].

Ref. [6] proposed an algorithm that discovers services using mobility-based clustering in MANETs. In order to maintain the stable cluster and select a suitable architecture, they proposed an algorithm for selecting the cluster head that takes into account the mobility of nodes and the network connectivity. While the mobility evaluation is based on direction, velocity, and duration, the connectivity evaluation is based on number of neighbors. Regarding the selection architecture, they proposed three architectures which are flooding, decentralized hash table, and centralized hash table. In order to find the most suitable architecture, the number of messages, the frequency of service requests, and the number of nodes in a cluster are required. The result showed that the proposed algorithm is more efficient when nodes have a specific movement pattern.

Ref. [7] presented a distributed service discovery architecture, called DSDM, for service sharing among diverse MANETs. The objective was to minimize the use of broadcasting used by other systems for service advertisement and discovery. In DSDM, servers register services with the closest directory, and that directory forwards the registration to all other directories in the network. A node selects its nearest directory for retrieving a service. DSDM also finds the most appropriate node to act as a directory. This directory selection algorithm considers capabilities of nodes, such as battery life, expected sensor lifetime, memory, and available bandwidth. Using these capabilities, a score is assigned to nodes, and those with the highest are selected as directories. When a directory goes offline, all nodes broadcast their scores and a new candidate is chosen. The results showed that as the number of directories increases, the average traffic (load) at each directory decreases, but at increasingly smaller rate. The disadvantage of DSDM is the requirement to know the capability of each device and the potentially high overhead due to continuous network wide broadcasting.

Ref. [10] proposed the Service Directory Placement Algorithm (SDPA). It uses a semi-Markov decision model to select the appropriate location of directories and a greedy algorithm for determining when to relocate. The results show that this algorithm is capable of reducing overhead by limiting the broadcasting of packets. The limitation is that the decision model requires significant knowledge of the network configuration and its complexity leads to it being run offline, rather than in a distributed manner during network operation.

Ref. [11] proposed a directory placement algorithm, as well as reduced power consumption by introducing a Basement Directory Agent (BDA). The BDA searches for a node which covers the most servers, and the directory is moved to that node. The result recommends the optimal number of directories and reduces the energy consumption as well.

Our research aims to improve the service discovery approach of [5] by applying the mobile directory mechanisms, which were developed by [10] to allow mobile directories relocated in WSNs. We propose an algorithm to select appropriate directories by using capability and connectivity (similar to [6] and [7]), and analyze the performance from the perspective of response time, successful request rate and communication overhead.

4. System Model and Design

In this section we present the assumed system model and propose the Service Discovery Relocation Algorithm (SDRA) that supports multiple, mobile directories.

4.1 System Model

We design SDRA for a moderate sized WSN with 50 to 300 nodes. Service discovery in small networks (10 to 30 nodes) is generally achievable without the complexity of directory agents. However as the number of nodes grow, this additional complexity can be outweighed by savings in performance. We have not yet considered very large networks (1000's of nodes) as they will be less likely to occur in practice and will probably require different mechanisms for service discovery (e.g. distributed hash tables). We assume that nodes in a WSN may be mobile (e.g. people, robots, and objects in factories) and are aware of their location (e.g. via GPS or other location technology). The scenario under consideration is a WSN for service discovery in a building, factory, or outdoor area.

We assume that sensors communicate using 6 Low PAN and IEEE 802.15.4, and run a WSN routing protocol so that shortest paths from source to destination can be found.

Wireless sensors are normally low capability devices, nonetheless, in some WSNs it is expected that a small percentage of nodes will have more capabilities than most sensors. For example, some nodes may be fixed and use an external power source, allowing for higher processing/storage capabilities and different wireless technologies (e.g. IEEE 802.11). Such nodes may be gateways to other networks, or act as dedicated relays for the network. When such nodes exist, it is possibly beneficial to locate directories on them as directories typically require more processing, storage and communications than clients and servers. In this paper we consider nodes with different capabilities. Relative to a normal sensor node we assume there are:

- N_B nodes which have a longer lasting battery. This capability is denoted as C_B, which is the normalized battery lifetime, i.e. for normal nodes, $C_B = 0$; for NB other nodes, $C_B = 1$.
- N_M nodes which have larger storage space, i.e. memory. This capability is denoted as C_M, which is the normalized size of memory, i.e. for normal nodes, $C_M = 0$; for N_M other nodes, $C_M = 1$
- N_W nodes which have dual wireless links, IEEE 802.15.4 and IEEE 802.11 (C_W).

As directories have to calculate and store more information than other nodes, and are expected to stay longer in the network, we take the capability of nodes into account when selecting a directory.

4.2 Overview of SDRA

In a resource-limited and de-centralized network such as a WSN, relying on centralized directories can create survivability problems (e.g. when directory nodes fail or leave the network) and performance bottlenecks, however multiple directories can improve the performance of service discovery. Therefore two design criteria must be considered: how many directory nodes are necessary and which nodes should host directories?

Regarding the number of directories allowed in a network, [7] considers and recommends less than 10% of the nodes to be directories. We follow this approach, setting an upper limit of 10% of all nodes to be available as a directory.

Selecting an undesirable node to be a directory can cause problems such as:

- Increased network communication overhead due to transfer of information between directories or registering services in case of registering to all directories.
- Increased processing and energy consumption on nodes due to many directories having to use more resources (e.g. respond to queries, retrieve relevant services).
- Increased cost of deploying many sufficient resources for being a directory.

Therefore selecting appropriate nodes to act as directories is important for service discovery performance. When using multiple directories for service discovery the two key challenges are:

- Network nodes determine which nodes should be directory node in a distributed manner.
- Clients (and servers) select which directory nodes they will send their request (or offer) to.

As the optimal placement of directories on nodes may vary over time, it is also important to allow directories to move between nodes, i.e. mobile directories. Our proposed Service Discovery Relocation Algorithm (SDRA) addresses these challenges. Fig. 2 illustrates the components of SDRA, which are described in the following sections.

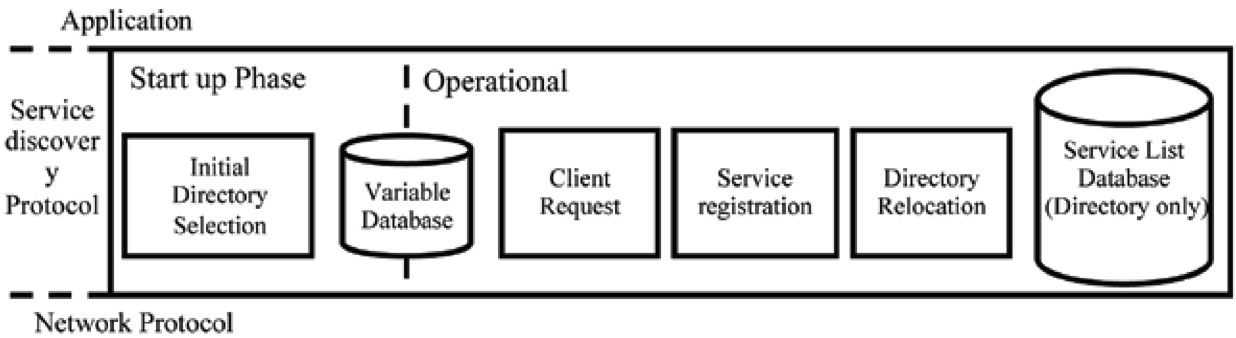

Fig. 2. System Components for SDRA.

4.3 Method of Selecting Directory Nodes

Using more directories can reduce the communication overhead and response time of service discovery [5, 7, 12]. According to our previous work [12], we have considered three algorithms for selecting the directory node.

The *most neighbor algorithm* selects the nodes with the most 1-hop neighbors to be directories. The idea is that with many clients as neighbors, the response time and overhead will be very small.

The *center-point algorithm* applies k-means clustering [14] to find center point nodes in the network which become the directories. Again, the idea is that with directories being the center point, many clients will be nearby, reducing response time and overhead.

A *random algorithm*, where nodes are randomly selected to be directories, is for comparing the result between the previous approaches. It is considered the baseline performance of service discovery.

The results in [12] show that using multiple mobile directory nodes performed better than using a single directory. However more directory nodes still lead to higher communication overhead and lower response time. The location of a directory is one factor that affects the performance of service discovery. For example, directories are sometimes located near each other (as opposed to spread across the network). This will lead to high response times and communication overhead for nodes which are far away. To overcome these issues and improve service discovery performance, in this paper, we offer a method for improving the selection of directory locations. There are several factors which can lead to the improvement of the service discovery performance. The first factor is the location of directories. Ideally directories should be located where they are reachable via as few hops as possible, from as many nodes as possible, i.e., where network density is high. The more clients close to a directory will lead to shorter response times and smaller overhead.

The next factor is the capability of a node. Directory nodes should have more resources than other nodes. Since the directory node has to store information about many services, it requires more memory than the other nodes. To avoid the lack of memory (and resulting in inability to store registered services), the node selected as the directory should preferably have a larger amount of memory.

Nodes with longer lasting battery are more suited to be a directory as they are likely to send/receive more packets and perform more processing (thereby consuming more energy) than other nodes. Also, with a larger battery a node has a longer lifetime in the network; this is beneficial for directories as they therefore will not leave the network as often as other nodes.

As a directory has to send/receive more packets than the other nodes, it will be beneficial if the directory node has a high wireless link capacity. Specifically, if a node has a second wireless link (e.g. IEEE 802.11) it can use that as a direct link with other directories.

The last factor is the distance between directory nodes. According to the previous work [12], in the most neighbor algorithm, a location of directory nodes which is too close will affect the communication overhead as well as the response time. Even using the maximum number of neighbors to select the directories, the neighbor of a directory and the next directory can be the same node; therefore the directories are not spread around the network.

Directory nodes should have sufficient resource for improving service discovery performance and also be located nearby other nodes. Therefore our SDRA considers nodes battery life (C_B), memory (C_M), wireless link (C_W) as well as the connectivity between nodes, i.e., the number of neighbors (N_{NB}).

To use the capability and connectivity information, each node must gather information about other nodes in the network. We assume that HELLO messages are exchanged between 1-hop neighbors - this is common in almost all wireless sensor networks - and therefore a node can easily learn about its own set of neighbors.

To gather information about other nodes, each node broadcasts its own information to the network. This information, called *NodeInfo* in Table 1, includes: node ID, set of 1-hop neighbors, node capabilities, node position. After a broadcasting period, each node receives information about all other nodes.

Table 1 Variable database in nodes.

Variable	Information
NodeInfo	ID, capability, position
NodeTable	NodeInfo for all nodes
NeighborList	List of 1-hop neighbors
DirectorySet	ID of existing directory nodes
DirectoryStatus	True if this node is a directory

A node stores information about all nodes in a *NodeTable*, with each row containing the ID, capabilities, position and its neighbors in *NeighborList*. Using their own *NodeInfo*, each node calculates a score (S) for all network nodes using equation (1), and then selects the ND nodes to be added to the DirectorySet with a condition that the distance between selected node must not be less than the transmission range (R) from any existing directory node in *DirectorySet*. The different capabilities can be assigned an importance weight by varying w_B, w_M and w_W.

$$S = N_{NB} + w_B C_B + w_M C_M + w_W C_W \tag{1}$$

The idea of the weights is to allow the network operator/users to prioritize the nodes selected as directories. For example, service discovery response time depends largely on the transmission capabilities of nodes. Therefore selecting nodes with larger transmission capabilities, i.e. high value of w_W, can potentially reduce response time. However another performance metric, the success rate of requests, may be reduced if directories do not have enough memory to store offers. Therefore selecting nodes with larger memory, i.e., high value of w_M, can potentially increase success rate. The weights should be chosen depending on the network scenario and usage requirements.

4.4 Relocation Method

A problem occurs when using a centralized architecture or single directory for service discovery when the directory leaves the network on its own accord or fails. To address this problem we propose a method to relocate a directory from one node to another. Where possible, a proactive relocation is used (e.g. when a node knows it will leave the network, Fig. 3(a)); otherwise a reactive relocation is performed (when other nodes discover a directory node has left the network, Fig. 3(b)).

When an existing directory determines it will leave the network soon (e.g. battery status goes below a pre-determined level), that node sends a message to its 1-hop neighbours, who then update their *NodeTable* and reply with their *NodeInfo* to the directory. The directory calculates the score using equation (1) in order to select a new directory from its set of 1-hop neighbours, and then informs them. The 1-hop neighbours add the new directory to their *DirectorySet* while removing the leaving node.

Although proactive relocation is preferred, sometimes a node may leave the network without notice. Similarly, due to node mobility, existing directory nodes may no longer be the optimal location of directories. Therefore a reactive relocation algorithm is also used. Every $T_{REACTIVE}$ seconds, each existing directory broadcasts a message to its 1-hop neighbors, who respond with their current *NodeInfo*. Then the existing directory selects a new directory if one of its 1-hop neighbors is more suitable. Note that this method only uses local (1-hop) broadcast, as opposed to broadcast to the entire network. The idea is that it is likely that a neighbor node would be suitable to be a directory due to similar connectivity. This scheme makes the tradeoff of reducing overhead, but at the expense of potentially sub-optimal location of directories.

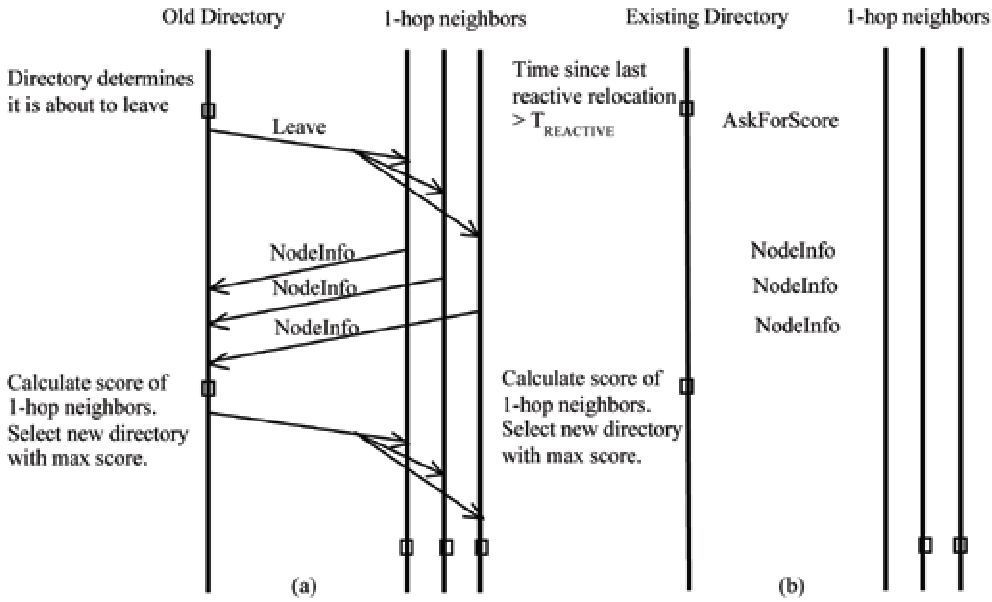

Fig. 3. System Components for SDRA.

4.5 Search Method

To improve the efficiency of a client searching for a server, SDRA uses a time to live method (TTL) in requests. The default behavior in service discovery (and in DSDM) is that if a client sends a request to its closest directory and the requested service is not available, the directory responds with a failure message. In SDRA, if the requested service is not available at the closest directory, that directory forwards the request to a neighbor directory. The number of directories that the request traverses is limited by the TTL set by the client. The TTL is decremented by each directory upon reception; a failure message is sent if the TTL reaches 0. The value of the TTL is set by the client and should be less than the maximum number of directories in the network. Fig. 4 shows that node 1 *(client)* initiates a request to find a relevant service, by sending a message to the closest directory (node 3). In this example, the requested service is not available, so the directory forwards the request to its closest directory (node 7). Then again, with the service unavailable, node 7 forwards the request to the directory on node 10. Assuming node 10 has the service, a response is sent back via the directories to the client node 1.

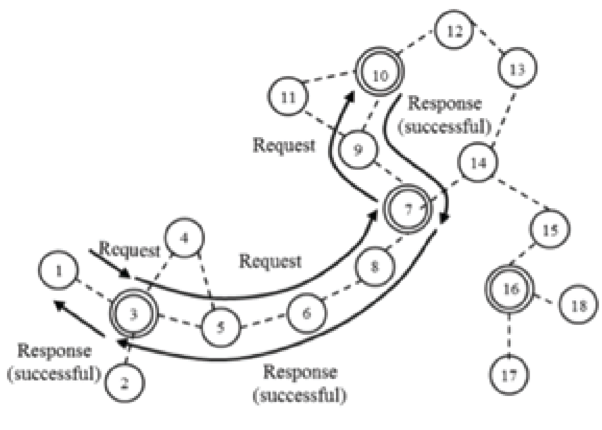

Fig. 4. Requests to multiple directories.

5. Simulation Model

5.1 Simulation Setup

We developed a simulation model in MATLAB to compare the performance of our proposed SDRA to DSDM [7]. The simulator models the network as a graph, where edges are created between vertices (nodes) when within range. Wireless link delay is constant (1 ms) and packets are sent along the shortest part between source and destination. We vary the number of directories (N_D), mobility (speed and pause time) and the number of services offered in the network. The parameters are listed in Table 2.

The weights for equation (1) were selected to give nodes with an extra wireless interface higher importance than other nodes, i.e. $w_W = 4$. Secondly, nodes with extra battery and extra memory are treated equally ($w_M = w_B = 2$). The ideal values of the weights will depend on the network scenario and node capabilities, and are subject to further study.

Table 2 System Parameters.

Number of nodes	
Total , N	100
Client, N_C	60
Server, N_S	40
Directory, N_D	1-10
Extra Memory, N_M	40
Extra Battery, N_B	40
Extra wireless, N_W	5
Mobility	
Model	Random Way Point
Speed	1-3 m/s
Pause time	1-10 s
Scenario	
Area	100 x 100 m
Transmission Range, R	15 m
Initial positions	Uniform distributed
Weight of Battery, w_B	2
Weight of Memory, w_M	2
Weight of Wireless link, w_W	4
Service and request	
Service type, $_{STYPE}$	10
Max Service/node, S_{MAX}	3
Request rate, λ	0.1 req/s, Poisson
Packet size, P	81 byte
Simulator	
Simulation duration	60 s
Seeds	25 seeds
Algorithm Parameters	
Interval for selecting directory, T_{SELECT_DA}	Start of simulation
Interval for reactive relocation, $T_{REACTIVE}$	Every 30 s.
Time to wait after all nodes broadcast, T_{WAIT}	1 s.

5.2 Performance Metrics

The following performance metrics are used to compare our SDRA against DSDM service discovery.

Successful Request Rate *(SRR)* is the percentage of all requests sent by clients and receiving a successful response. A request is unsuccessful if either no response is received (e.g. the directory cannot be reached) or a negative response is received (e.g. the service request is not available).

Response Time *(RT)* is the time from when a client initiates a request until it receives a successful response. As the requests/responses are relatively small packets, the response time depends largely on the path taken, and in particular the number of hops in that path. For our analysis we assume the packet delay for each hop is identical.

Communication Overhead *(CO)* is the sum of all bytes sent in order to perform the service discovery process. We count the bytes sent over each hop as overhead, e.g. sending a 100 Bytes packet over 4 hops contributes an overhead of 400 Bytes. We separate the communication overhead to that from three phases:

- Sending request: the overhead of sending request to directory until getting a response back from directory
- Registering service: the overhead of registering a service to a directory.
- Directory management: the overhead of selecting directory and also relocating to a new directory.

6. Results

Key results are presented in Figs. 5-8. The data points are the average across 25 simulations, each with different random seeds. The 95% confidence interval is also shown as error bars (in those plots with no error bars, the confidence interval was very small and hence removed for clarity). Results for other scenarios have also been obtained, but show similar trends to those presented in Fig. 5-8.

Fig. 5 shows that both SDRA and DSDM have almost 100% successful request rate. With increasing directories, the result should be getting better because directories are located throughout the whole network, thus it will enhance the opportunity of a reachable request from client (request node) to directory.

The more directories there are, a lower average response time is expected. The reason is if directories are spread all over the network, clients and servers will take a short time for communicating with directories. To expect the results of DSDM should be better than SDRA (the average response time should be less than SDRA) because in DSDM each directory contains all services (whereas in SDRA services are only registered to one directory).

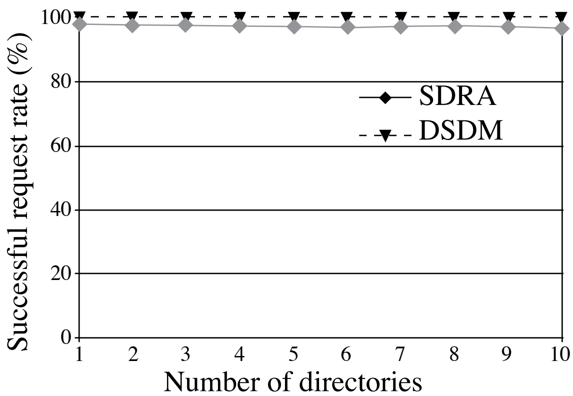

Fig. 5. Successful request rate.

Therefore, the client will be able to find the service from the first directory. However, Fig. 6 shows that the average response time of DSDM is higher than SDRA. This is because the location of directories selected by DSDM is not spread throughout the whole network as expected. We demonstrate this by using the average number of hops in SDRA and DSDM as shown in Fig. 7. It shows that the number of hops between clients and the closest directory of SDRA is smaller than DSDM, i.e., SDRA does a better job of distributing the directories across the network.

The communication overhead is an important criterion to indicate the impact of service discovery in WSNs. Fig. 8(a-d) show results on different contributors to overhead. The overhead of sending a request by clients in Fig. 8(a) illustrates that the communication overhead reduces, as the number of directories increase.

This figure also shows that SDRA can operate more efficiently than DSDM due to the location of directories in DSDM as mentioned above. Therefore clients take more time for sending requests to a directory.

Fig. 8(b) shows the overhead of registering services. With SDRA, the overhead is quite low, and reduces as the number of directories increase since providers become closer to directories. A key advantage of SDRA compared to DSDM can be seen in Fig. 8(b). In DSDM a large overhead is incurred due to servers registering their service with multiple (all) directories. SDRA makes the trade-off of reducing this overhead, at the expense of request time for clients to find directories being potentially higher. But as shown previously in Fig. 6 and 7, the response time is reduced in SDRA scheme by improved placement of directories and allowing search via multiple directories.

SDRA has a slightly larger relocation overhead (Fig. 8(c)) than DSDM because it performs proactive and reactive relocations. However, this contributes little to the total overhead.

Fig. 8(d) shows the total overhead of SDRA and DSDM. In SDRA when the number of directories becomes higher, the overhead is reduced and vice versa in DSDM. The overhead of registering services has a high impact on the total overhead.

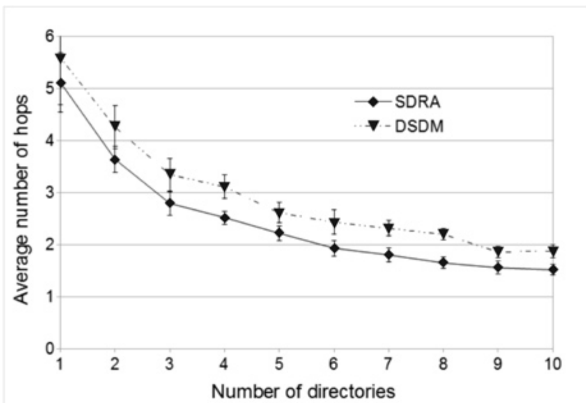

Fig. 6. Average number of hops.

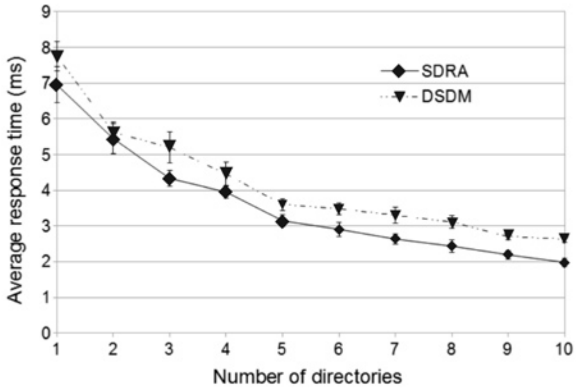

Fig. 7. Average response time.

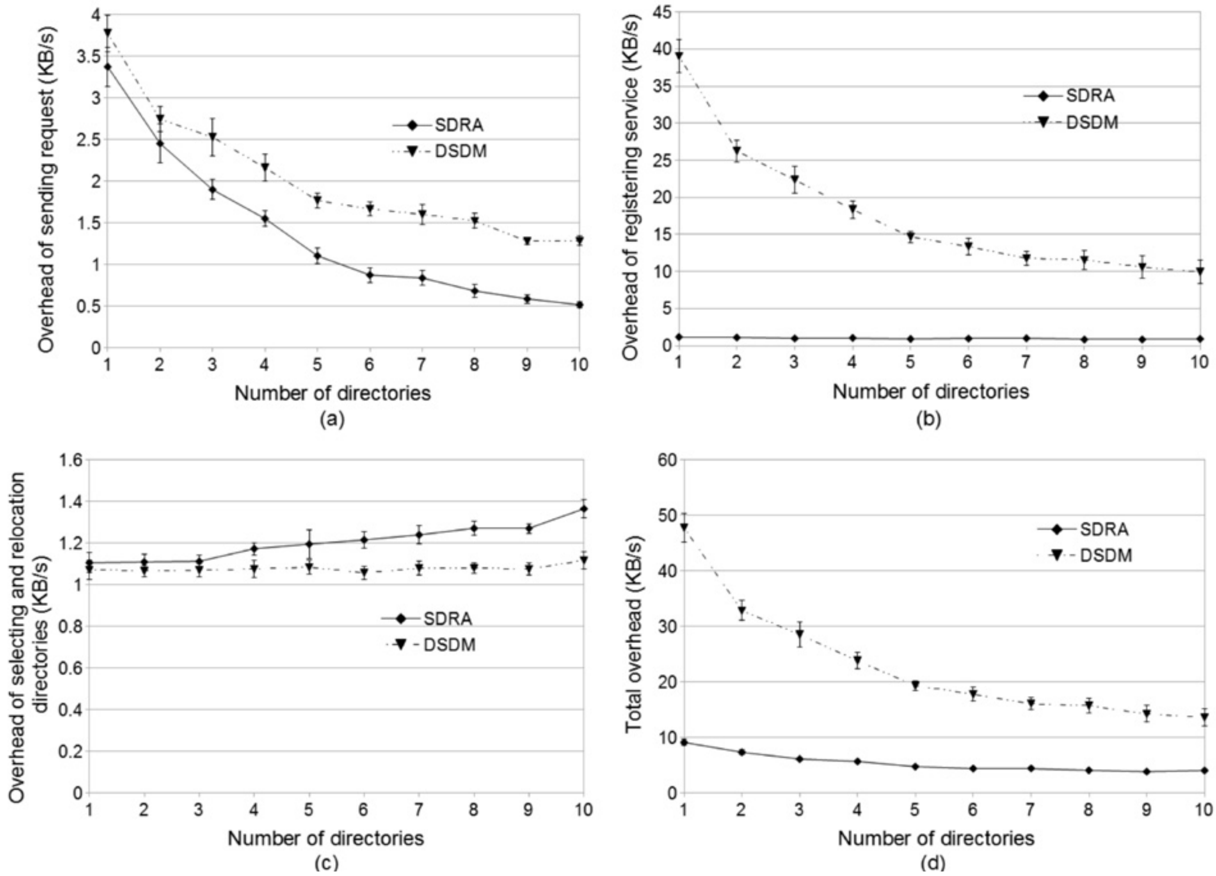

Fig. 8. Communication overhead: (a) client requests; (b) service registration; (c) directory management; (d) total.

7. Conclusion

We have proposed SDRA, an algorithm for selecting and relocating multiple, mobile directories to improve service discovery in WSNs. SDRA consists of:

- An algorithm for selecting directories that takes the capability of nodes into account as well as the connectivity.
- Use of a time to live in client requests, allowing them to be forwarded to multiple directories.
- Methods to relocate a directory to a new node when a directory leaves the network or goes offline.

In comparison with a similar approach, DSDM, our SDRA introduces lower total communication overhead primarily by reducing the service registrations sent from servers to directories. Rather than registering with all directories, in SDRA registration is only to the closest directory. SDRA maintains a successful request rate by allowing forwarding of requests between directories. The location and mobility of directories allows for lower response time compared to DSDM.

In addition to analyzing more complex scenarios, future work will enhance the proactive and reactive relocation to allow for better placement of directories upon node failure.

8. Acknowledgement

This research is financially supported by Thailand Advanced Institute of Science and Technology, National Science and Technology Development Agency, Tokyo Institute of Technology, and Sirindhorn International Institute of Technology, Thammasat University. This work was also supported by the National Research University Project of Thailand Office of Higher Education Commission.

9. References

[1] S.A Munir, X. Dongliang, C. Canfeng, J. Ma., Service Discovery Wireless Sensor Networks: Protocols & Classifications, In Proc. 11th Intl. Conf. Advanced Communication Technology, pp. 1007 - 1011, 3 Apr. 2009.

[2] F. Zhu, M.W Mutka, L.M. Ni., Service Discovery in Pervasive Computing Environments, In Proc. IEEE PerCom, pp. 81 - 90, 21 Nov. 2005.

[3] W. Tian, G.S. Kuo., An Analytical Model for Centralized Service Discovery Architecture in Wireless Networks, In Proc. IEEE VTC, pp. 1- 5, 28 Sep. 2006.

[4] P. Wei, L. Xi Cheng., On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks, In Proc. ACM MobiHoc, pp. 129 - 130, 6 Aug. 2002.

[5] S.A. Chaudhry, S.H. Chauhdary, D.J. Won, H.A Ali, K.H Kim., A Proxy-Enabled Service Discovery Architecture to Find Proximity-Based Services in 6LoWPAN, In Proc. Intl. Conf. Embedded Ubiquitous Computing, pp. 956 - 965, 2006.

[6] Y.S. Shim, Y.S. Kim, K.H. Lee., A Mobility-Based Clustering and Discover of Web Services in Mobile Networks, In Proc. Int. Conf. Web Services, pp. 374 - 380, 31 Jul. 2009.

[7] H. Artail, K.W. Mershad, H. Hamze., DSDM: A Distributed Service Discovery Model for Manets, In IEEE Trans. Parallel and Distributed Systems, Vol.19, No. 9, pp. 1224 - 1236, 2008.

[8] F.M. Anwar, M.T. Raza, Y. Seung-Wha, K. Ki-Hyung., ENUM Based Service Discovery Architecture for 6LoWPAN, In Proc. IEEE WCNC, pp. 1525 - 3511, 8 Jul. 2010.

[9] S. Siva, D. Niranjan, G. Srividya, V. Sridhar., A Distributed Service Discovery Architecture for Pervasive Environments, In Proc. Joint Intl. Conf. Autonomic and Autonomous Systems and Intl. Conf. Networking and Services, pp. 25, 2005.

[10] G.V. Sergio, T.V. Son, C.M. Leung., A Mobile-Directory Approach to Service Discovery in Wireless Ad Hoc Networks, In IEEE Trans. Mobile Computing, Vol. 7, No. 10, pp. 1242 - 1256, 2008.

[11] M. TaeKwon, B. Zafar, R. JaeTek, K. Ki Hyung, Y. SeungWha., The Effective DA Placement Method in SLP, In Proc. Intl. Conf. Advanced Communication Technology, pp. 364 - 368, 10 Feb. 2010.

[12] K. Thanatporn, S. Gordon., Performance of Using Multiple Mobile Directories for Service Discovery in Wireless Sensor Networks, In Proc. 3rd Intl. Conf. Information Communication Technology for Embedded Systems, 5 pp. 22 - 24 Mar. 2012.

[13] M.J. Kim, M. Kumar, B.A. Shirazi., Service Discovery Using Volunteer Nodes for Pervasive Environments, In Proc. 5th Intl. Conf. Pervasive Services, pp. 188 - 197, 11 - 14 Jul. 2005.

[14] F. Zhu, M.W. Mutka, L.M. Ni., Service Discovery in Pervasive Computing Environments, In IEEE Pervasive Computing, Vol. 4, No. 4, pp. 81 - 90, 2005.

[15] C.N Ververidis, G.C Polyzos., Service Discovery for Mobile Ad Hoc Networks: a Survey of Issues and Techniques, In IEEE Communication Surveys & Tutorials, Vol. 10, No. 3, pp. 30 - 45, 2008.

[16] L. Ting, J. Yang, X. Peng, Z. Chen, L. Chengyang., Semi-supervised Learning Algorithm Based on Simplified Association Rules Combining with K-mean and Its Application in Land Evaluation, In Proc. 4th Intl. Conf. Natural Computation, pp. 316 - 320, 7 Nov. 2008.

[17] G. Tao, K.Hung, K.Y. Jian., Towards a Flexible Service Discovery, J. Network and Computer Application, Vol. 28, No. 3, pp. 316 - 320, 2005.