ASDP: An Action-Based Service Discovery Protocol Using Ant Colony Algorithm in Wireless Sensor Networks

Hongwei Huo, Deyun Gao, Yanchao Niu, and Shuai Gao

School of Electronics and Information Engineering, Beijing Jiaotong University,
Beijing, 100044, China
hwhuo@ieee.org, gniux819@gmail.com, {gaody,shgao}@bjtu.edu.cn

Abstract. In large-scale wireless sensor networks, efficient service discovery and data transmission mechanisms are both essential and challenging. Ant colony algorithm which has been used to resolve routing, localization and object tracing issues in mobile ad hoc and sensor networks provide a valuable solution for this problem. In this paper, we describe a novel scalable Action-based Service Discovery Protocol (ASDP) using ant colony algorithm in wireless sensor networks. ADSP can abstract the semantics information from the data via the nodes or user operation and map them into six different action sets. Then it adjusts the related parameters to satisfy with service and transmission requirements from different kinds of actions. We evaluate it against other approaches to identify its merits and limitations. The simulation results show that ASDP can maximize the network utilization. Farther experiments indicate it scales to large number of nodes.

1 Introduction

Typically, services are applications offered by service providers which users can interact with. A service discovery protocol allows service providers to advertise their services, and also allows users to automatically discover such services [4][5]. Wireless Sensor Networks (WSNs), as an emerging technology, open a wide perspective for future applications in ubiquitous computing. It is not unpredictable that large-scale sensor networks eventually will become one of the most important infrastructures in our life, which can be integrated into homes, offices, public places, battle fields and all kinds of natural environments, where service discovery will be revered significantly. The concept of service can be extended as either hardware or software resources, accessible through the network and capable of interacting with each other by sending and receiving information and data. And the service discovery can be comprehended as nodes are aware of their own capabilities, automatically locate network services according to the semantic attributes and to advertise their own capabilities to the rest of the network, by which they can cooperate with other nodes in the network, for the purpose of providing networking and system services such as user querying or data recording [2] [3].

For efficient service discovery, the main challenges in wireless sensor networks are the limited power, computational capacities, memory and prone to failures. Some

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 338–349, 2007.
© Springer-Verlag Berlin Heidelberg 2007

applications involve mobility and large number of nodes. In WSNs, the network actions consists of the task sent to the nodes and the data recoded by nodes, thus communication is application-specific and data-centric. Typical approaches for service discovery and followed data transmission rely on either flooding or centralized storage. Both may cause efficiency, scalability and robustness problems. In this paper, we describe a novel scalable, efficient and robust Action-based Service Discovery Protocol-ASDP using Ant colony algorithm, which has been widely used to resolve routing, localization and object tracing problems in mobile ad hoc and sensor networks, to maximize the network utilization. Considering the special requirement of services discovery in wireless sensor networks, we select the basic application scenarios, which consist of user query and data recording, and create a scheme to abstract the semantics information from the data generation via the nodes or user operations via the sinks. All these information are mapped in to six different basic network action sets: emergency querying, emergency report, normal querying, normal report, emergency cooperation and normal cooperation. ASDP selects different parameter combination of the ant-based algorithm to achieve the service requirement separately. We ran extensive simulations to investigate our approach, and evaluated it against other approaches to identify its merits and limitations. The results show that ASDP is scalable to large number of nodes and is highly efficient with action's changes.

The rest of this paper is organized as follows. Section 2 describes the service discovery framework in wireless sensor networks. In section 3, we present an action based ant colony optimization algorithm which is used as the soul of our work. Section 4 explains our service discovery protocol design, followed in Section 5 by simulation and performance evaluation results. We discuss related work in section 6. Finally, in section 7 we give out some conclusions.

2 Action-Based Service Discovery in WSNs

We consider the solutions for service discovery depend on the application-specific and data-centric network action, which consists of the task sent to nodes and the data recoded by nodes[5][6]. According to different user requirements, Service Discovery protocol should perform different actions: service query, service match and service reply, which will perform different schemes. In the basic application scenarios of WSNs, the sensor nodes are deployed randomly or regularly to perform sensing, caching, processing, and wireless transmission of sensor data via self-organizing. Users can pre-configure the sensor nodes to record sensory data and to report them periodically or according to the user's request[10]. Figure.1 depicts the 3 basic service discovery types in detail. In this paper, we abstract this process into six basic actions: emergency querying (EQ), emergency report(ER), normal querying(NQ), normal report(NR), emergency cooperation(EC) and normal cooperation(NC). Emergency querying is an abstraction of the querying by users via one or more sink nodes to obtain data from WSNs within an extremely short time, while normal querying is much like emergency querying but without time limitation. Emergency report means that the nodes should request the location of the sink nodes and report some emergency data to these nodes according to the pre-configuration. Normal report means that the nodes report regular data to these nodes periodic or on demand. To facilitate the cooperation of sensor nodes to support the reconfiguration capability or to achieve a common goal of the network, sensor nodes should also request services from each other, thus we define another two basic actions: emergency cooperation for real-time scenarios and normal cooperation for non-emergency scenarios. A good example is the monitoring of emergency or disaster scenarios, such as floods, fires, earthquakes, and landslides. Sensor nodes used to collect and transmit data can fail suddenly and unpredictably as they may melt, corrode or get smashed. This sudden, unpredictable failure is especially troubling because of the potential loss of data collected by the sensor nodes. Thus data backup between sensor nodes is an important mission for this kind of WSNs. The nodes should request the data backup service to other nodes with potential storage space[11]. This process is a kind of emergency cooperation action.

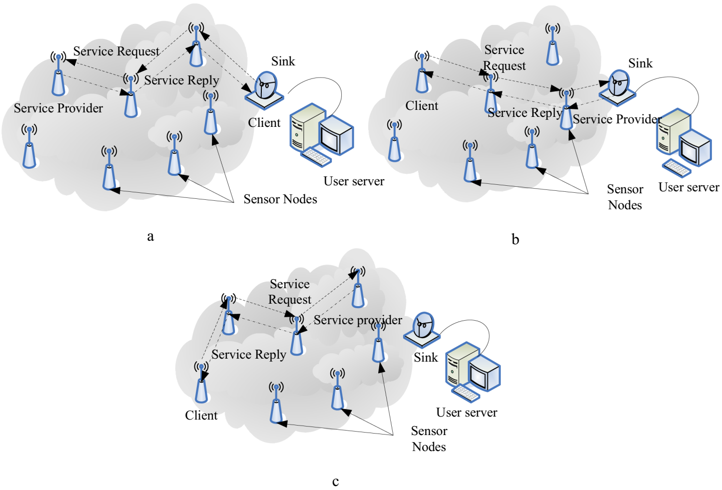

Fig. 1. Three basic service discovery : a)sink nodes as client and sensor nodes as provider. b) sensor nodes as client and sink nodes as provider. c) both client and provider are sensor nodes.

All the services in WSNs should consult one or more of these actions and assimilate them as their properties according to the semantics information. In fact how to describe the service and how to map semantics information into network action is out of our scope. In this paper we mainly pay attention to how to fulfill the communication request of the service discovery in each action. However, we also give a brief introduction. Figure.2 summarizes the approach to map the semantics information into basic actions. We hold a service parser as an interface to parse the semantics information into different attributes, such as session origination, session destination, time constraints, energy efficiency, etc… When a session from top layer is received, service parser will check these attributes and classify this session into its basic action to fulfill the requirement of the session.

Service discovery might be affected by architecture, available resources, incorporated routing and communication mechanisms[1].But typical approaches for service discovery and followed data transmission rely on either flooding or centralized storage, which may cause efficiency, scalability and robustness problems. These approaches can not fulfill the request of all actions of the network concurrently. Many biological systems exhibit properties of self-organization and emergence, which are inherently adaptive, but there is no centralized control. Clearly such systems are somewhat parallel with WSNs, which likewise lack central controllers and must adapt to prevailing conditions[6].

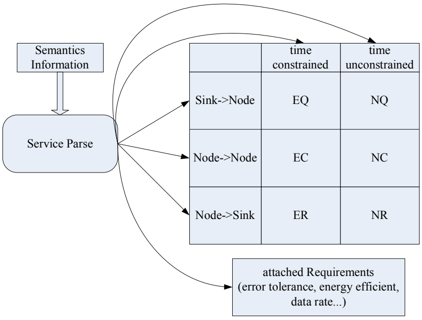

Fig. 2. Approach to map the semantics information into basic actions

3 Ant Colony Optimization Algorithm

The ant can seek path between the nest and multiple food sources. They accomplish the mission with great efficiency. When the environment changes, ants can also quickly discover new routes. Since Service Discovery is a process to find the best service and steer the service request from the client to the service provider, and then transmit the response back to the client. In dynamic and stochastic networks, such as WSNs, to search the best service and establish a proper route between them is a NP-hard Problem[14]. The idea of ASDP is inspired by the process used by ant colony to discover food resources, in which individual ants are capable of finding the best or relative Optimized path to the food sources or the nest without global administration. It is desirable for our Service discovery protocols to achieve optimization performance for all actions. In this section, we will introduce the Action-based Ant Colony Optimization Algorithm in detail.

3.1 Assumptions and Definitions

We assume that the network is represented by a weighted graph $G=(E,V)$, where V is the set of nodes and E is the set of edges. In WSNs, since sensor nodes may join or fail randomly, G can be considered to be a dynamic graph. Let Si denotes the set of Sink nodes, thus $N=V-Si$ denotes the set of sensor nodes. If node i and node j can communication directly via wireless radio, the link between *i* and *j* can be represented by $e^{i,j}$,where, $e^{i,j}\in E$. We give some assumptions and definitions:

In this paper, we assume that 1) all the communication links are symmetric, despite the link are asymmetrical in many practical communication systems. In fact, the symmetrical link can be assured through the MAC layer protocol.2) the local clock of the node is synchronized or approximately synchronized. This is very important, because the delay between nodes need to calculate via local clock.3) all of sensor nodes here have the same initial energy, denoted as E_0 .4) the localization coordinates of sensor nodes are known and the nodes were located at the same horizontal surface. Thus, we can use the 2-D Euclidean distance formula to calculate the distance of two sensor nodes. This assumption is crucially useful in simulation experiments.

Definition 1. Service Requirement. *The demand for services of each network action is different。 To facilitate the description of our algorithm, we define two basic* ***Service Requirements*** *according to the different network actions :* ***Delay*** *and* ***Potential Energy***。

Delay. *We use* $D_L^{i,j}$ *to represent the end to end delay between two nodes i and j. For a service S, we denote the path between the Client (denoted as 1) and Service Provider (denoted as m)* $P(S)$ *.Thus the delay for a service path is* $D_L(P(S)) = \sum_{i=1}^{m-1} D_L^{i,i+1}$.

Potential Energy. *We use* $E_D^{i,j}$ *represent the energy consumption caused by the communication between nodes i and j .For* $P(S)$ *which is defined above, the* ***energy consumption*** *of service path(****Potential Energy****) is* $E_D(P(\overline{S})) = \sum_{i=1}^{m} E_D^{i,i+1}$.

Definition 2. Target Value Function. *In the case of EQ, EC and ER, the we mainly consider delay, rather than energy. Whereas, in the case of NQ, NC and NR, we mainly consider energy, and delay is not an obvious concern. Therefore, we can set and adjust the action chosen parameter* $\vec{\gamma}$ *neatly according to the semantics describe of service request to reflect the specific service requirement. Where,* $\vec{\gamma}$ *is a 2-D vector,* $\vec{\gamma} = [\gamma_D, \gamma_E]$, $\gamma_D, \gamma_E \in [0,1]$ *.And* $\vec{\gamma}$ *can also be seen as the weight parameters of the* ***Service Requirement***

For a Service Discovery process, we define a Target Value Function, f :

$$f(P(S)) = \vec{\gamma} \times \begin{bmatrix} \omega_1 \times (\overline{D_{\text{threshold}}} / 2 - D_L(P(S))) \\ \omega_2 \times (|P(s)| E_0 - E_D(P(S))) \end{bmatrix} \tag{1}$$

For each Service, we always choose the highest value of f as the best service and the path of this service as the service reply route. Where, $\overline{D_{\text{threshold}}}$ represent the Maximum allowable delay of the delay threshold. n. $|P(s)| = m$ represents the total hop count of the path.

At this point, the problems of action based service discovery can be described as: for a service S, try to find a subset of E and one of its edge sets (a subset of V) in graph G to create a path set P(S), which makes numerical value of $f(P(S))$ as great as possible.

3.2 Basic Procedures

As mentioned above, the ant colony optimization algorithms have been inspired by the behavior of the real ant colony. In order to find the optimum solution, we generate several artificial ants, which search the solution space. The probabilistic movement of the ant allows the ant to explore new paths and find the proper service provider. We define three types of ants in this algorithm: forward ant(***Fa***), backward ant(***Ba***), and service ant(***Sa***). ***Fa*** is generated from the client to explore a path to a proper service provider. ***Ba*** is generated from the service provider and route back to the client. ***Sa*** is also generated from service provider to publish its services. High level flow chart for the action based ant colony optimization algorithm is depicted in Fig.3.

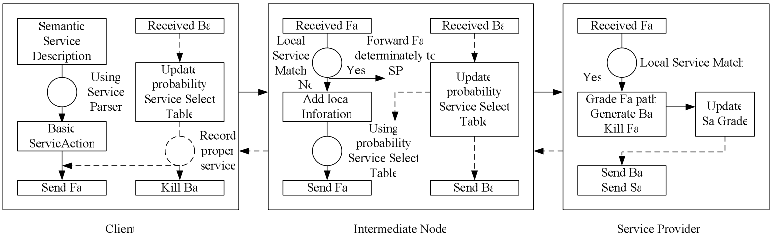

Fig. 3. High level flow chart describing basic procedures

When a client (either node or a sink) in the WSNs wants to find a service and /or maintain a path from the client to the service provider, it sends Fa to search for the proper service provider. For the service request is parsed from Semantic Service Description, ***Fa*** carried the action chosen parameter mapped via Basic Action to adjust the service discovery process. The generation rate of the ***Fa***s are determined by the network dynamics and the required ability to quickly respond to changes of the WSN. A Fa moves in WSNs to search the service provider using the Probability Service Select Table (PSST). PSST is maintained in each node according to which, the ***Fa*** probabilistically selects the next hop in its searching path. Figure.4 shows the structure of the PSST, where $\mathrm{Pr}_{i,k,j}$ (can also be marked as $\mathrm{Pr}_{k,j}$ in some obvious scenarios) denotes the probability of current ***Fa*** to achieve the Service Provider j via the next hop k which is a neighbor of current node i. N_i denotes the number of the valuable neighbors of current Node i, and N_{sp} is the number of the valuable service providers. $\mathrm{Pr}_{i,k,j}$ is calculated as following:

$$\mathrm{Pr}_{i,k,j} = \begin{cases} \dfrac{(\tau_{i,k})^{\alpha}(\eta_{i,k})^{\beta}}{\sum_{u \in N_i}(\tau_{i,u})^{\alpha}(\eta_{i,u})^{\beta}}, v_k \notin tabu(Fa) \\ 0 \qquad\qquad\qquad\qquad , v_k \in tabu(Fa) \end{cases} \tag{2}$$

Where, *tabu()* is used to represent the set of nodes that the ant has already passed. $\tau_{i,k}$ represents the current pheromone value on link $e^{i,k}$,which will be refreshed by

Bas. $\eta_{i,k}$ is the local heuristic gene, which represent the expectation to choose a proper neighbor. Here, we define $\eta_{i,k}$ as following:

$$\eta_{i,k} = \gamma_D \omega_1 (1 - {}^{2D_L^{i,k}}\!/_{D_{threshold}}) + \gamma_E \omega_2 ({}^{E_D^{i,k}}\!/_{E_0}) \tag{3}$$

Thus, $\eta_{i,k}$ lay out that we always expect to choose the neighbor with high Energy Potential and low Delay links. α and β are two parameters to reflect the importance of current pheromone value and the local heuristic gene.

$\mathrm{Pr}_{1,1}$	$\mathrm{Pr}_{1,2}$	$\mathrm{Pr}_{1,3}$	• • •	$\mathrm{Pr}_{1,N_{sp}}$
$\mathrm{Pr}_{2,1}$	$\mathrm{Pr}_{2,2}$	$\mathrm{Pr}_{2,3}$	• • •	$\mathrm{Pr}_{2,N_{sp}}$
• • •	• • •	• • •	• • •	• • •
$\mathrm{Pr}_{N_i,1}$	$\mathrm{Pr}_{N_i,2}$	$\mathrm{Pr}_{N_i,3}$	• • •	$\mathrm{Pr}_{N_i,N_{sp}}$

Fig. 4. Probability Service Select Table of Node i

Fas collect the paths' information and intermediate nodes' local information as they travel. When a node received a ***Fa***, it first checks if the service requested by the ***Fa*** exists in its local service cache. If there is no local service match, ***Fa*** will be forwarded according current node's PSST. If there is local service match, the current node will differentiate whether the service is provided by itself or other nodes. If the service is not provided by itself, the node will forward the ***Fa*** to the Service provider determinately. Otherwise, it will grade the path information carried by the ***Fa*** and record this information. Then the ***Fa*** will be killed and a Ba will be generated and sent in the reverse path of its corresponding ***Fa*** with the grade φ, all ids of the intermediate nodes visited by ***Fa***. φ is the parameter which represent the ability of a service provider. Here, we define $\varphi = {}^{E_{sp}}\!/_{E_0}$, E_{sp} represents the current energy of service provider. That means the energy condition is the main factors to be considered for the ability of service provider.

As ***Fa*** move in the reverse path, the intermediate nodes will modify their pheromone value for the corresponding service and update the PSST. Here, $\tau_{i,k}$ will be updated according to the global information carried by the ***Ba***. When a ***Ba*** is received, the intermediate node collects the grade φ and calculate new $\tau_{i,k}$, using $\tau_{i,k}(n+1) \leftarrow \varphi \times \tau_{i,k}(n) + \Delta\tau_{i,k}$. Where,

$$\Delta\tau_{i,k} = \frac{\gamma_D \omega_0 (\overline{D_{threshold}} / 2 - D_L(P(s))) + \gamma_E \omega_1 (|P(s)| E_0 - E_D(P(s)))}{|P(s)|} \tag{4}$$

Finally, the client will receive the Ba, update its PSST and record the proper service. Then the Ba will be killed. We defined the initialized $\tau_{i,k}$ and $\Delta\tau_{i,k}$ randomly.

4 Special Consideration for ASDP Design

We noted that in massive sensor networks, the size of ***Sa*** and ***Ba*** will extend significantly. The ant will be so big that it may unfeasible to send the ant through the network. In this scenario, we can modify our packet format using the approach proposed in [15] and just maintain 3-most current hops in our ants, and the rest information can be stored in intermediate nodes.

When a service is deployed, it usually sends its service description to nearby nodes, which can increase the chance of its discovery by a matching service request. In our algorithms we use ***Sas*** to achieve this goal. When a service is deployed, it doesn't send any ***Sas*** until the first service request arrives. When the first service request for this service is arrived, a ***Sa*** with service description will be sent to all its neighbors with the value of TTL set as 1. If the Service provider received n service request from different clients during a fixed time interval Δt , it will send another ***Sa*** with a value of TTL n-1 at the end of this time interval. If the received service request came from the same client, it will send another ***Sa*** with a value of TTL 1. Else, it will not send anything about this service. In each neighbor of this service provider, the ***Sa*** will be killed and the information will be deleted if it can not receive a new ***Sa*** in a fixed time interval Δt. Then during the time interval, if another ***Sa*** with TTL more than 1 is received, it will update the ***Sa*** information and forward the ***Sa*** to its other neighbors with the TTL decreased by 1. This scheme means a popular service will have more chance of being discovered than a less popular one.

5 Simulation and Performance Evaluation

We implemented the ASDP using C++, and verified the performance of the proposed protocol through extensive simulation experiment. For executing comparison, we also implemented Flooding and HESED [22] in the same environment.

We first evaluate the Average Time Delay to match a service and Average Energy Consumption to match a service, using ASDP, HESED and Random Search(RSD) separately. In this pattern, we deployed only one service node in an $1000m \times 300m$ area. We increased the number of nodes from 20 to 100 to evaluate the performance of ASDP in small WSNs. The communication radius of them are 200m. In all experiments, the $D_{\text{threshold}} = 5s$. And we balance delay and energy consumption here. Figure.5 and Figure.6 illustrate the result of this experiment. We also evaluated the performance of ASDP in large scale WSNs. Figure.7, 8 show the results of this experiment. The results show that the delay of ASDP and HESED are nearly same, but ASDP is more energy efficient. Though RSP is more energy efficient than ASDP and HESED, the long time delay may be insufferable for most of the applications.

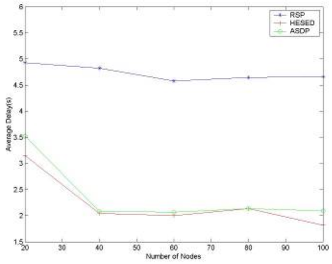

Fig. 5. Average Delay in small WSNs

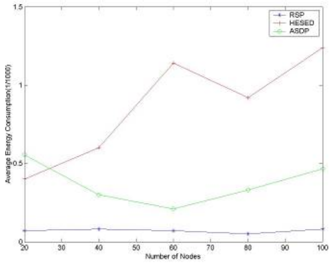

Fig. 6. Energy Consumption in small WSNs

As action based Service discovery protocol, we had to evaluate ASDP in different actions. Firstly, we imagine that there are more than one service in our WSNs such as NQ or EQ. We deployed 10 service nodes randomly and tried to match the best service in 5 seconds. We compared the response of HESED and ASDP. Figure.9 and Figure.10 show the total time and total energy to match the best service. The results indicate that HESED can match the best service quickly with more energy consumption, while ASDP has the opposite result. We take notice of the process of matching the best service, ASDP can finish this distributed rather than determined by client itself.

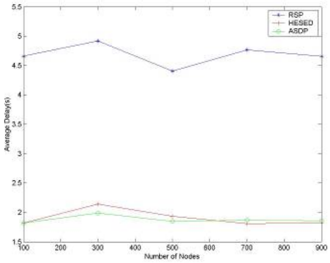

Fig. 7. Average Delay in Large Scale WSNs

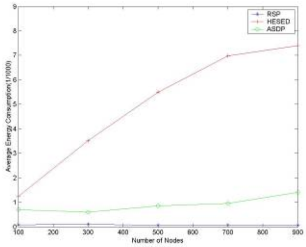

Fig. 8. Energy Consumption in large scale WSNs

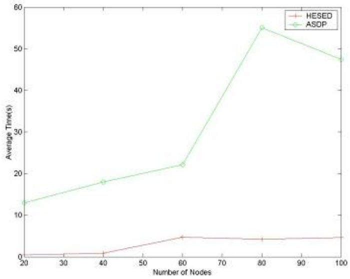

Fig. 9. Time of the best service matching

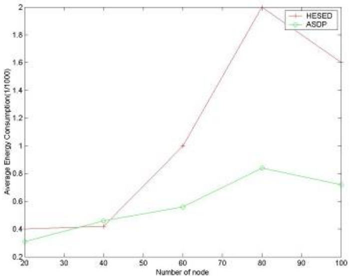

Fig. 10. Energy of the best service matching

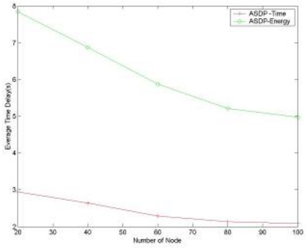

Fig. 11. Average Delay of different actions

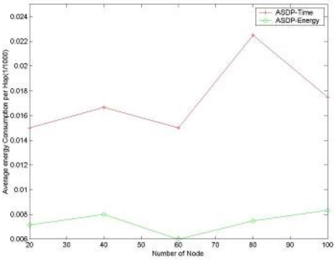

Fig. 12. Energy consumption of different actions

Secondly, we evaluate the affection of the weight parameters of the Service Requirement to ASDP. We set $\vec{\gamma}_1 = [50,0.5]$ and $\vec{\gamma}_2 = [0.5,50]$ to evaluate the action of ASDP. $\vec{\gamma}_1$ considers delay is more important and $\vec{\gamma}_2$ considers energy is more important. We showed the result of time delay of each action in figure.11.We compared the energy consumption of each hop in different actions, too. The result is shown in Figure 12. Both Fig.11 and Fig.12 show that ASDP is more sensitive and flexible with action's changes.

6 Related Work

In WSNs, the simplest form of service discovery is global flooding, which does not scale well. Many organizations have designed and developed service discovery protocols[16]. These still have not been mature enough to be used by industry for mobile ad hoc and sensor networks environment. However, many research works on Service discovery in MANET and WSNs have been conducted over the past few years. In [17], Dipanjan Chakraborty et al. introduced group based service discovery protocol which considered little mobility and energy efficiency. Yu Yang et al consider multicast-based SDPs may be more suitable for MANETs. They proposed HESED in [22]. Katsigiannis, C.O. et al. also proposed an architecture for reliable service discovery and delivery in MANETs based on power management employing SLP extensions. [18] Jui Chi Liang, Mobile Service Discovery Protocol (MSDP)for Mobile Ad-Hoc Networks .Marin-Perianu, proposed an Energy-Efficient Cluster-Based Service Discovery in WSNs via topology control scheme in [2][10]. Sethom. K. and Afifi. H also introduced a simple service discovery architecture for sensor networks[8].

Ant Colony Optimaization approach is proposed by Marco Dorigo, and has been used to resolve routing, localization and object tracing problems in mobile ad hoc and sensor networks. L.Zhang et al. proposed an ant colony based multicasting protocol in MANET to minimize the route overhead in [12]. In [15], Tiago Camilo proposed two kinds of energy-efficient ant colony based routing algorithms for WSNs, and compared the performance of these algorithms. In [20][21], Zheng X.Q and Liu L.G tried to use an ant colony based algorithm to design QoS-Aware Routing Algorithm for MANETs.Ricky Robinson et.al hold the same view with us and considered that ant colony and WSNs have many common characteristics. They were the first to propose a scheme to use ant colony approach to execute service discovery[6]. However, they just tried to use ant colony in the Service Advertisements and did not carry out a detailed evaluation.

7 Conclusions

WSNs have already opened a wide perspective for future applications especially in ubiquitous computing. Thus we identified service discovery as one of the major design components. Inspired by ant colony, we described a novel scalable Action-based Service Discovery Protocol (ASDP) using ant colony algorithm in WSNs in this paper. ADSP can abstract the semantics information from the data generation via the nodes or user's operation via the sinks and map them in to six different action sets. Then it adjusts the related parameters to satisfy with service and transmission requirements from different kinds of actions. We evaluated it against other approaches to identify its merits and limitations. The simulation results show that ASDP can maximize the network utilization. The farther experiments indicate it scales to large number of nodes. In the future, we plan to further improve the performance of ASDP under different movement, node failure or different traffic scenarios and try to summarize approach to adjust the parameters.

Acknowledgements

We would like to thank the NSFC for their financial support of the grant project (No. 60473001) Hongwei Huo would like to thank Beijing Jiaotong University for their financial support of the Ph.D Student Innovation Fund Program (No.48022).

References

1. Zhu, F., Mutka, M.W., Ni, L.M.: A Private, Secure, and User-Centric Information Exposure Model for Service Discovery Protocols. IEEE Transactions on Mobile Computing 5(4), 418–429 (2006)
2. Marin-Perianu, R., Scholten, H., Havinga, P., Hartel, P.: Energy-Efficient Cluster-Based Service Discovery in Wireless Sensor Networks. In: LCN 2006. Proceedings of 31st IEEE Conference on Local Computer Networks, pp. 931–938 (November 2006)
3. Katsigiannis, C.O., Kateros, D.A., Koutsoloukas, E.A., Tselikas, N.-L.D., Venieris, I.S.: Architecture for Reliable Service Discovery and Delivery in MANETs based on Power Management Employing SLP Extensions. IEEE Wireless Communications 13(5), 90–95 (2006)
4. Tyan, J., Mahmoud, Q.H.: A Comprehensive Service Discovery Solution for Mobile Ad Hoc Networks. Mobile Networks and Applications 10(4), 423–434 (2005)
5. Liljana, G., Ramjee, P.: Ad Hoc Networking Towards Seamless Communications, 1st edn. Ch. 6, pp. 143–172. Springer, Netherland (2006)
6. Robinson, R., Indulska, J.: A Complex Systems Approach to Service Discovery. In: Galindo, F., Takizawa, M., Traunmüller, R. (eds.) DEXA 2004. LNCS, vol. 3180, pp. 657–661. Springer, Heidelberg (2004)

7. Zhang, Y., Cao, J., Chan, A.T.S., Chan, K.C.C.: Sensors and Wireless Sensor Networks for Pervasive Computing Applications. Journal of Ubiqutous Computing and Inteligence 1(1), 17–34 (2007)
8. Sethom, K., Afifi, H.: A New Service Discovery Architecture for Sensor Networks. In: WTS 2005. Proceedings of Wireless Telecommunications Symposium, 2005, pp. 190–196 (April 2005)
9. Xinlian, Z., Mi, W.: Service Discovery Protocol in Wireless Sensor Networks. In: SKG 2006. Proceedings of the Second International Conference on Semantics, Knowledge, and Grid, pp. 101–102 (November 2006)
10. Marin-Perianu, R.S., Scholten, J., Havinga, P.J.M., Hartel, P.H.: Cluster-based Service Discovery for Heterogeneous Wireless Sensor Networks, Technical Report: TR-CTIT-07-05, University of Twente, Netherlands (May 2007)
11. Kamra, A., Misra, V., Feldman, J., Rubenstein, D.: Growth Codes: Maximizing Sensor Network Data Persistence. In: Sigcomm 2006. Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 255–266 (September 2006)
12. Zhang, L., Shen, D., Shan, X., Li, V.O.K.: An Ant-based Multicasting Procotol in Mobile Ad-Hoc Network. International Journal of Computational Intelligence and Applications 5(2), 185–199 (2005)
13. Marin-Perianu, M., Hofmeijer, T.J., Havinga, P.J.M.: Assisting Business Processes through Wireless Sensor Networks. In: ICT 2006. Proceedings of 13th International Conference on Telecommunications, pp. 6–10 (May 2006)
14. Dorigo, M., Stutzle, T.: Ant Colony Optimaization. Ch. 4, pp. 215–238. The MIT press, Cambridge (2004)
15. Camilo, T., Carreto, C., Silva, J.S., Boavida, F.: An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) ANTS 2006. LNCS, vol. 4150, Springer, Heidelberg (2006)
16. Zhu, F., Mutka, M.W., Ni, L.M.: Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing, 97–112 (December 2005)
17. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Transaction on Mobile Computing 5(2), 97–112 (2006)
18. Liang, J.C., Chen, J.C., Zhang, T.: Mobile Service Discovery Protocol (MSDP) for Mobile Ad-Hoc Networks. In: ISADS 07. Proceedings of 8th International Symposium on Autonomous Decentralized Systems, pp. 352–362 (April 2007)
19. Hussein, O., Saadawi, T.: Ant routing algorithm for mobile ad-hoc networks (ARAMA). In: IPCCC.2003. Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference, pp. 281–290 (April 2003)
20. Liu, L., Feng, G.: A Novel Ant Colony Based QoS-Aware Routing Algorithm for MANETs. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 457–466. Springer, Heidelberg (2005)
21. Xiangquan, Z., Lijia, G., Wei, G., Renting, L.: A cross-layer design and ant-colony optimization based load-balancing routing protocol for ad-hoc networks. Frontiers of Electrical and Electronic Engineering in China 2(2), 219–229 (2007)
22. Yang, Y., Hassanein, H., Mawji, A.: A New Approach to Service Discovery in Wireless Mobile Ad Hoc Networks. In: ICC 2006. Proceeding of 2006 IEEE International Conference on Communication, pp. 3838–3843 (June 2006)