Secure Service Discovery Protocols for Ad Hoc Networks

Haitham Elwahsh, Mohamed Hashem, and Mohamed Amin

King Saud University, Ain shams University, Minoufiya University
Haitham.elwahsh@gmail.com, {mhashem100,mohamed_amin110}@yahoo.com

Abstract. Ad-hoc networks, mobile devices communicate via wireless links without the aid of any fixed networking infrastructure. These devices must be able to discover services dynamically and share them safely, taking into account ad-hoc networks requirements such as limited processing and communication power, decentralized management, and dynamic network topology, among others. Legacy solutions fail in addressing these requirements. In this paper, we propose a service discovery protocol with security features, the Secure Pervasive Discovery Protocol. SPDP is a fully distributed protocol in which services offered by devices can be discovered by others, without a central server. It is based on an anarchy trust model, which provides location of trusted services, as well as protection of confidential information, secure communications, or access control.

Keywords: ad-hoc networks, service discovery protocol, security.

1 Introduction

Recent advances in microelectronic and wireless technologies have fostered the proliferation of small devices with limited communication and processing power. They are what are known as "pervasive systems". Personal Digital Assistants (PDAs) and mobile phones are the more "visible" of these kinds of devices, but there are many others that surround us, unobserved. For example, today most household appliances have embedded microprocessors. Each one of these small devices offers a specific service to the user, but thanks to their capacity for communication, in the near future they will be able to collaborate with each other to build up more complex services. In order to achieve this, devices in such "ad-hoc" networks should dynamically discover and share services between them when they are close enough. In ad-hoc networks composed of limited devices, it is very important to minimize the total number of transmissions, in order to reduce battery consumption of the devices. It is also important to implement mechanisms to detect, as soon as possible, both the availability and unavailability of services produced when a device joins or leaves the network. Security in these networks is also critical because there are many chances of misuse both from fraudulent servers and from misbehaving clients. In this paper, we propose a service discovery protocol with security features, the Secure Pervasive Discovery Protocol (SPDP). SPDP is a fully distributed protocol in which services offered by devices can be discovered by others, without a central server. It provides location of trusted services, as well as protection of confidential information, secure communications, identification

N. Meghanathan et al. (Eds.): CCSIT 2011, Part I, CCIS 131, pp. 147–157, 2011.

between devices, or access control, by forming a reliable ad-hoc network. The paper is organized as follows: section 2 enumerates the main service discovery protocols proposed so far in the literature, we will see that none of them adapts well to ad-hoc networks. Section 3 presents our secure pervasive discovery protocol, SPDP, with its application scenario, and description of the algorithm. In section 4 we describe the underlying trust model as security support. In section 5 we present the simulation results comparing SPDP with other services discovery protocols. Finally, we conclude with some conclusions and future work.

2 Related Works

Dynamic service discovery is not a new problem. There are several solutions proposed for fixed networks, with different levels of acceptance, like SLP [RFC2608, 1999] [1], Jini [Sun, 1999] [4] and Salutation [Miller and Pascoe, 2000] [5]. More recently, other service discovery protocols, specifically designed for ad-hoc networks, have been defined, some tied to a wireless technology (SDP for Bluetooth [SDP, 2001] [6], IAS for IrDA [IrDA, 1996]) [7], others that jointly deal with the problems of ad-hoc routing and service discovery (GSD [Chakraborty et al., 2002] [8], HSID [Oh et al., 2004]) [9], and others that work at the application layer of the protocol stack (DEAPspace [Nidd, 2001] [12], Konark [Helal et al., 2003] [13], and the post-query strategies [Barbeau and Kranakis, 2003]) [14]. Only a few protocols have built-in security, the most important are SSDS [Czerwinski et al., 1999] [16]and Splendor [Zhu et al., 2003]. However, these solutions cannot be directly applied to an ad-hoc network, because they were designed for and are more suitable for (fixed) wired networks.We see three main problems in the solutions enumerated: – First, many of them use a central server, such as SLP2, Jini and Salutation. It maintains the directory of services in the network and it is also a reliable entity upon which the security of the system is based. An ad-hoc network cannot rely upon having any single device permanently present in order to act as central server, and furthermore, maybe none of the devices present at any moment may be suitable to act as the server. – Secondly, the solutions that may work without a central server, like SSDP, are designed without considering the power constraints typical in wireless networks. They make an extensive use of multicast or broadcast transmissions which are almost costless in wired networks but are power hungry in wireless networks. – Thirdly, security issues are not well covered. SSDS provides security in enterprise environments but may not work in ad-hoc networks with mobile services. Splendor does not provide certificate revocation and trust models of PKIs. They both depend on trustworthy servers and they propose solutions which are provided at the IP level. Accepting that alternatives to the centralised approach are required, we consider two alternative approaches for distributing service announcements: – The “Push” solution, in which a device that offers a service sends unsolicited advertisements, and the other devices listen to these advertisements selecting those services they are interested in. – The “Pull” solution, in which a device requests a service when it needs it, and devices that offer that service answer the request, perhaps with third devices taking note of the reply for future use. In ad-hoc networks, it is very important to minimise the total number of transmissions, in order to reduce battery consumption. It is also important to implement

mechanisms to detect as soon as possible both the availability and unavailability of services produced when a device joins or leaves the network. These factors must be taken into account when selecting between a push solution and a pull solution. The DEAPspace algorithm is the only service discovery protocol, listed above, that tries to minimize the total number of transmissions. It uses a pure "push" solution and each device periodically broadcast its "world view" although none of them has to request a service.

3 SPDP: Secure Pervasive Discovery Protocol

In this paper we propose a new service discovery protocol, the Secure Pervasive Discovery Protocol (SPDP), which merges characteristics of both pull and push solutions to improve the performance of the protocol. Also, SPDP provides security based on an anarchy trust management model. Such trust management model does not require neither a central trusted server nor a hierarchical architecture, so it is suitable to overcome the challenges imposed by ad-hoc networks such as no central management, no strict security policies and highly dynamic nature (see section 4). The Secure Pervasive Discovery Protocol (SPDP) is intended to solve the problem of enumerating the services available in ad-hoc networks, composed of devices with limited transmission power, memory, processing power, etc. Legacy service discovery protocols use a centralized server that listens for broadcast or multicast announcements of available services at a known port address, and lists the relevant services in response to enquiries. The protocol we propose does away with the need for the central server. Ad-hoc networks cannot rely upon having any single device permanently present in order to act as central server, and further, none of the devices present at any moment may be suitable to act as the server. One of the key objectives of the SPDP is to minimize battery use in all devices. This means that the number of transmissions necessary to discover services should be reduced as much as possible. A device announces its services only when other devices request the service. Service announcements are broadcasted to all the devices in the network, all of which will get to know about the new service simultaneously at that moment, without having to actively query for it. In addition, SPDP allows sharing services safely, through an underlying trust management model between devices, which allows us to store service information from other "alleged" trusted service agents and later to use them if such information is really authentic and pright. Currently, the security support provided by service discovery protocols are focused on authentication, integrity, and confidentiality [RFC2608, 1999] [Czerwinski et al., 1999] [Zhu et al., 2003]. Even more, some of them include authorization services as part of the discovery [Zhu et al., 2003]. Such support is based on IPSec [Kent and Atkinson, 1998] or traditional PKI in the last case. However, these security services could be not necessary for the discovery, but they could cause energy and processing consumption. Protecting both energy and processing consumption is a very essential issue for devices with limited capabilities. So we have considered providing basic security services to prevent certain attacks (i.e. DoS, false announcements, and false services) and to avoid the sending of unnecessary messages. In the remainder of this section, we present the application scenario for SPDP and some considerations to be taken into account. Then, we will formally describe the algorithm used to implement it.

3.1 Application Scenario

Let's assume that there is an ad-hoc network, composed of D devices, each device offers S services, and expects to remain available in this network for T seconds. This time T is previously configured in the device, depending on its mobility characteristics. Each device has an SPDP User Agent (SPDP UA) and an SPDP Service Agent (SPDP SA). The SPDP UA is a process working on the user's behalf to search information about services offered in the network. The Service Agent SPDP (SPDP SA) is a process working to advertise services offered by the device. The SPDP SA always includes the availability time T of its device in its announcements. Each device has a cache associated which contains a list of the services that have been heard from the network. Each element e of the cache associated to the SPDP UA has three fields: the service description, the service lifetime and the service expiration time. The service expiration time is the time it is estimated the service will remain available. This time is calculated as the minimum of two values: the time the device has promised to remain available, and the time the server announced that the service would remain available. Entries remove themselves from the cache when their timeout elapses. With regard to security, each device handles a list of reliable devices and the trust degree associated with them. Trust helps devices to limit their cache size; services from untrusted devices are not stored in the cache. Depending on the trust degree, a device decides to store the service offered by a device on its cache. When the devices access services, devices with biggest trust degree are selected in the first place.

3.2 Algorithm Description

The SPDP has two mandatory messages: SPDP Service Request, which is used to send service announcements and SPDP Service Reply, which is used to answer a SPDP Service Request, announcing available services. SPDP has one optional message: SPDP Service Deregister, which is used to inform that a service is no longer available. Now, we will explain in detail how SPDP UA and SPDP SA use these primitives.

3.2.1 SPDP User Agent

When an application or the final user of the device needs a service of a certain type, it calls its SPDP UA. In order to support different application needs, in SPDP we have defined two kinds of queries:

– **one query–one response** (1/1): the application is interested in the service, not in which device offers it.

– **one query–multiple responses** (1/n): the application wants to discover all devices in the network offering the service. In this kind of query, we introduce a special type of service, named ALL, in order to allow an application to discover all available services of all types in the network.

3.2.2 SPSP Service Agent

The SPDP SA advertises services offered by the device. It has to process SPDP Service Request messages and to generate the corresponding SPDP Service Reply, if necessary. In order to minimize the number of transmissions, the SPDP SA takes into account the type of query made by the remote SPDP UA.

4 Evaluating the SPDP Protocol

In this section we present a performance evaluation study of SPDP in a ubiquitous computing environment. We compare our protocol with the theoretical distributed approaches, push and pull; because all the service discovery protocols defined in the literature are based on one of these approaches; and also we compare PDP with the service discovery protocol standard in Internet, SLP, and with UPnP's SSDP. This study was carried out through simulation using the well-known network simulator,NS-2. Our simulator is available in [Campo and Perea, 2004]. During the simulation, devices join the ubiquitous environment at random times, request and offer random services, and leave the network after a random time. The number of devices in the network varies over time, but its mean remains stationary. Random times follow exponential distributions, while random services follow uniform distributions. For simplicity we assume that each device offers just one service. The parameters of the simulation are**: the mean number of devices**, **the mean time they remain available in the network**, **the size of the caches**, **the mean time between service requests**, and **the total number of service types**. The results of interests are: the number of messages (the number of messages transmitted in the network normalized to the number of service request), the service discovery ratio (the ratio of services discovered to the total number of services available in the network) and the error ratio (the ratio of services discovered that were not available in the network to the total number of services discovered). Figure 6 shows the number of messages transmitted, the service discovery ratio and the error ratio, in a scenario with 20 devices, an average device life time ranging from 600 to 19200 seconds, a cache size of 100 entries, 5 different types of services, and each device requesting a random service every 60 seconds. The SPDP number of messages is quite under those obtained for SLP and for pull solutions, while keeping the same service discovery ratio and error rate of them.

(Fig. 2) shows the global power consumption in the same scenario as before. We see that, despite of using broadcast transmissions instead of unicast, and despite of sending bigger service requests (with the services already known from the cache), PDP achieves an important reduction in the power consumed, that is a reflection of the reduction of messages transmitted, and also of the lower energy cost associated with receiving than with transmitting. Only SSDP with an announcement period of "60 s" achieves less consumption, but at the cost of a higher error and lower service discovery percentages. Furthermore, as we will show later, PDP preserves the battery of the more limited devices, while the other protocols equally deplete the batteries of all devices. Regarding the delay in service discovery, it depends on the way the service discovery is done. In push mode protocols, the answer is obtained immediately from the cache. In directory-based protocols, the delay is the associated with transmitting a service request message to the directory, the processing time the directory needs to obtain the answer from its services database, and the transmission time associated with sending the reply. In pull mode protocols, as well as in Multicast DNS and PDP, the device broadcasts a service request (perhaps consulting first its local cache), and then it must wait for answers to come during a given period of time.

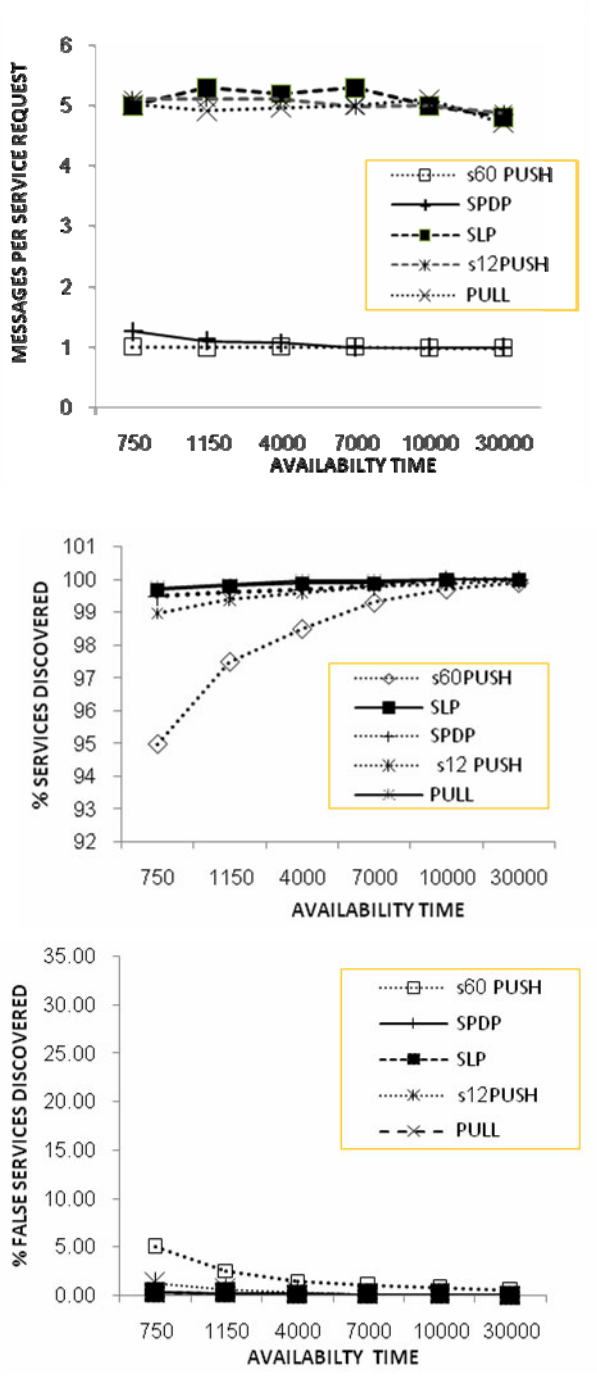

Fig. 1. Comparison of SPDP with others protocols

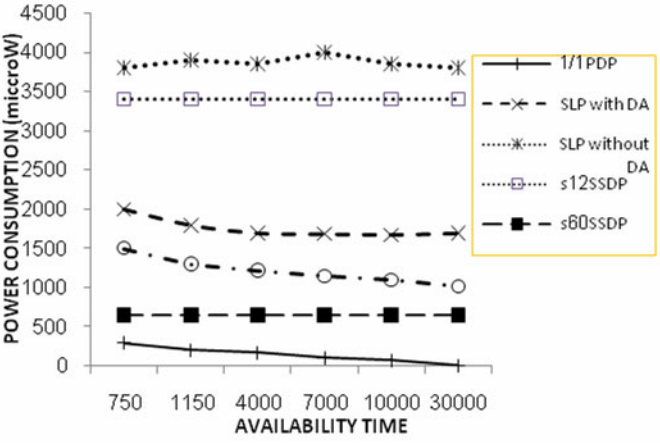

Fig. 2. Comparison of power consumption

Now, we will study the impact of the number of devices in the network and the cache size in the performance of PDP. A PDP with cache 0 is equivalent to a pull mode.(Fig. 3) shows that if the cache size is big enough, the number of messages transmitted remains constant, since all the services are already known and stored in the cache. For small cache sizes, when the number of devices equals the cache size, the number of messages starts growing linearly. For cache 0 (pull mode) the increment is always linear. Now, we will demonstrate how PDP takes into account device heterogeneity, achieving a reduction of traffic transmission (and so power consumption) in the more limited devices.(Fig. 4) shows the percentage of replies sent by each kind of devices depending on its availability time. We have considered a scenario with 40 devices in mean, with five different availability times: 500, 2500, 4500, 6500 and 9500 s, with about 20% of devices (in mean) of each type. The rest of parameters of the simulation are the same as before, except that the cache size for devices with availability time 500 and 2500 is 10 services, while for devices with availability time 4500 and 6500 is 40 services and for devices with availability time 9500 is 100 services. This way we simulate that devices that move more frequently (PDAs, mobile phones) have less memory than devices that move less frequently (laptops or desktop computers). In (Fig. 4) we see that devices with greater availability time answer more requests, preserving power consumption of devices with smaller availability in the figure sum up 70%, because in PDP some requests generate no replies (all known services were already included in the request). Considering this, devices with availability of 9500 s answer almost 50% of the service requests. If other service discovery protocol were used, all devices would answer with equal probability, 20%. This means that with PDP, fixed devices with greater availability time and less limitations answer most of the requests. This was one of the objectives of our protocol.(Fig. 4) also shows that devices with very small availability time (in our case, 500 s) answer more requests than devices with middle availability times. This is because these devices are highly mobile, continually change of networks, and in each new environment they arrive, they have to answer requests above their own services, to

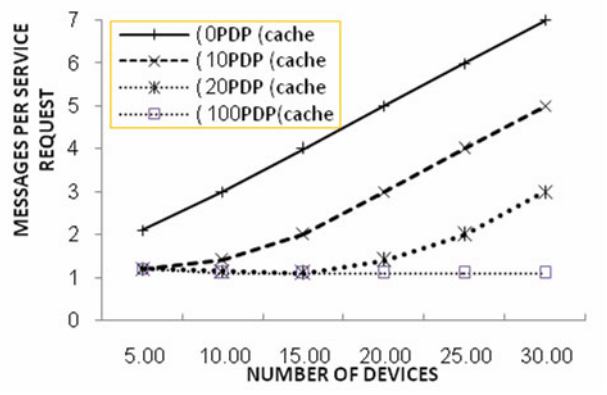

Fig. 3. Service replies per search for different number of devices

make them known to the rest of the devices. As we know, PDP is a fully distributed protocol, it does not rely in any central directory. However, with this simulation we show that PDP is designed time. It is worth mentioning that all percents shown in such a way that, if there are devices that are less mobile (remain more time in the environment) and that have more memory, most of the queries will be answered by them, relieving the more mobile and limited ones of answering, and so preserving their battery. In this scenario we assume that devices with higher availability time also have greater cache's sizes. This is a realistic assumption, since fixed devices use to have more memory than mobile (small, battery powered) devices All the above figures considered PDP one query– multiple responses queries. If the application is interested in the service, not in which device offers it, PDP one query–one response (1/1) can be used instead, obtaining a further reduction in number of messages and power consumption. (Fig. 5) compares PDP one query–one response against the same service discovery protocols as before.

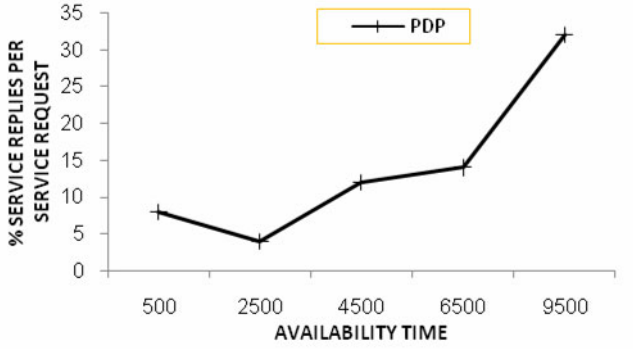

Fig. 4. Service replies per search in an heterogeneous environment with PDP

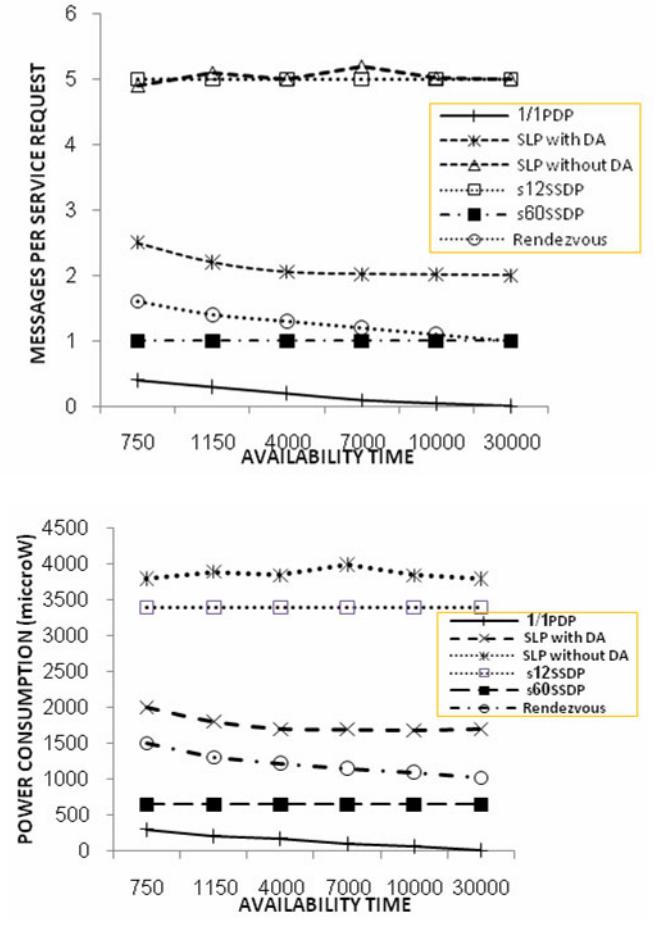

Fig. 5. Comparison of PDP 1/1 with other protocols

5 Conclusions

Ad-hoc networks are becoming increasingly common thanks to the development of mobile device technology. When a device connects to an ad-hoc network, it wants to know the services offered by the network and in turn it may offer its own services. Client applications in the device want to discover trustworthy services automatically, while server applications want to be used by trustworthy clients that will not misuse or attack them. Additionally, secure network communication is also an important issue. These goals are carried out by SPDP. SPDP is a suitable service discovery protocol for ad-hoc networks since:

– It is based on a distributed open architecture; therefore, it does not require central servers;

– It has a simple architecture which contains only two types of components, user agents and service agents;

– It provides autonomous and mobile agents with a simple method for discovering services that are available;

– It minimizes battery use in all devices since the number of transmissions necessary to discover a service is reduced as much as possible;

– It integrates a security model in order to guarantee the security level required by devices. Security issues include authenticity, and data integrity based on a decentralized trust management model.

References

[1] Guttman, E., Perkins, C., Veizades, J., Day, M.: RFC 2608: Service Location Protocol, Version 2 (June 1999)

[2] Goland, Y.Y., Cai, T., Leach, P., Gu, Y.: Simple service discovery protocol/1.0. Internet-draft (work in progress), draft-cai-ssdp-v1-03.txt (April 1999)

[3] Cheshire, S.: DNS-Based Service Discovery. Internetdraft (work in progress) (February 2004)

[4] Jini Architectural Overview, White Paper (1999)

[5] Salutation Consortium (1998), http://www.salutation.org

[6] Bluetooth Specification v1.1, Part E: Service Discovery Protocol (SDP)

[7] Infrared Data Association, Infrared data association link management 1.1 (January 1996)

[8] Chakraborty, D., Joshi, A., Yesha, Y., Fini, T.: GSD: A novel group-based service discovery protocol for MANETS. In: 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden (September 2002)

[9] Oh, C.-S., Ko, Y.-B., Kim, J.-H.: A hybrid service discovery for improving robustness in mobile ad hoc networks. In: The International Conference on Dependable Systems and Networks, DSN 2004, Florence, Italy (July 2004)

[10] Koodli, R., Perkins, C.E.: Service discovery in ondemand ad hoc networks (draft-koodli-manetservicediscovery- 00.txt), Internet-draft (work in progress) (October 2002)

[11] Engelstad, P.E., Zheng, Y.: Evaluation of service discovery architectures for mobile ad hoc networks. In: 2nd Annual Conference on Wireless On-demand Network Systems and Services (WONS 2005), pp. 2–15 (January 2005)

[12] Nidd, M.: Service discovery in DEAPspace. IEEE Personal Communications (August. 2001)

[13] Helal, S., Desai, N., Verma, V.: Konark—A service discovery and delivery protocol for ad-hoc networks. In: Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans (March 2003)

[14] Barbeau, M., Kranakis, E.: Modeling and performance analysis of service discovery strategies in ad hoc networks. In: International Conference on Wireless Networks, ICWN 2003, Nevada, Canada (June 2003)

[15] Apple, Rendezvous (2004), http://developer.apple.com/macosx/rendezvous/

[16] Czerwinski, S.E., Zhao, B.Y., Hodes, T.D., Joseph, A.D., Katz, R.H.: An architecture for a secure service discovery service. In: Proc. Mobicom 1999 (1999)

[17] Helal, S.: Standards for service discovery and delivery. IEEE Pervasive Computing, 95–100 (July/September 2002)

[18] Toh, C.-K.: Ad Hoc Mobile Wireless Networks. In: Protocols and Systems, Prentice-Hall PTR, Englewood Cliffs (2002)
[19] Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: IEEE INFOCOM (2001)
[20] Morais Cordeiro, C., Gossain, H., Agrawal, D.P.: Multicast over wireless mobile ad hoc networks: present and future directions. IEEE Network 17(1), 52–59 (2003)
[21] Pascoe, B.: Salutation-lite. Find-and-bind technologies for mobile devices, Technical Report, Salutation Consortium (June 1999)
[22] Kaminsky, A.: JiniME: Jini connection technology for mobile devices. Technical Report, Information Technology Laboratory, Rochester Institute of Technology (August 2000)
[23] Li, G.: JXTA: A network programming environment, IEEE Internet Computing, 88–95 (May–June 2001)
[24] Dobrev, P., Famolari, D., Kurzke, C., Miller, B.A.: Device and service discovery in home networks with OSGi. IEEE Communications Magazine, 86–92 (August 2002)
[25] Grimm, R., Davis, J., Lemar, E., Macbeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S., Wetherall, D.: Programming for pervasive computing environments, Technical Report, University of Washington (June 2001)
[26] Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., Lilley, J.: The design and implementation of an intentional naming system. In: 17th ACM Symposium on Operating Systems Principles (SOSP1999), pp. 186–201 (December 1999)
[27] Cheshire, S.: Performing DNS queries via IP Multicast, Internet-draft (work in progress) (February 2004)
[28] Esibov, L., Adoba, B., Thaler, D.: Linklocal Multicast Name Resolution (LLMNR), Internet-draft (work in progress) (July 2004)
[29] Helal, S., Desai, N., Verma, V., Arslan, B.: Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. IEEE Transactions on Systems, Man, and Cybernetics 33(6), 682–696 (2003)
[30] Campo, C.: Service discovery in pervasive multi-agent systems. In: Finin, T., Maamar, Z. (eds.) AAMAS Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Agents, Bologna, Italy (July 2002)
[31] Campo, C., Munõz, M., Perea, J.C., Mariń, A., García- Rubio, C.: GSDL and PDP: a new service discovery middlewareto support spontaneous interactions in pervasive systems. In: Middleware Support for Pervasive Computing (PerWare 2005) at the 3rd Conference on Pervasive Computing, PerCom 2005 (March 2005)
[32] Siva Ram Murthy, C., Manoj, B.S.: Ad Hoc Wireless Networks: Architectures and Protocols. Prentice-Hall PTR, Englewood Cliffs (2004)
[33] Meyer, D.: RFC 2365: Administratively Scoped IP Multicast (July 1998)
[34] Campo, C., Perea, J.C.: Implementation of pervasive discovery protocol (2004), `http://www.it.uc3m.es/celeste/pdp/`
[35] Guttman, E., Perkins, C., Kempf, J.: RFC 2609: Service Templates and Service: Schemes (June 1999)