

# Service and Device Discovery of Nodes in a Wireless Sensor Network

Åke Östmark, Per Lindgren

Dept. of Computer Science and Electrical Engineering  
Luleå University of Technology, Sweden  
ake.ostmark@csee.ltu.se  
per.lindgren@csee.ltu.se

Aart van Halteren, Lianne Meppelink

Faculty of Electrical Engineering, Mathematics and  
Computer Science  
University of Twente, The Netherlands  
a.t.vanhalteren@utwente.nl  
l.meppelink@student.utwente.nl

**Abstract**—Emerging wireless communication standards and more capable sensors and actuators have pushed further development of wireless sensor networks. Deploying a large number of sensor nodes requires a high-level framework enabling the devices to present themselves and the resources they hold. The device and the resources can be described as services, and in this paper, we review a number of well-known service discovery protocols. Bonjour stands out with its auto-configuration, distributed architecture, and sharing of resources. We also present a lightweight implementation in order to demonstrate that an emerging standards-based device and service discovery protocol can actually be deployed on small wireless sensor nodes.

**Index Terms:** Service discovery, wireless sensor, sensor networks.

## I. INTRODUCTION

Sensors are an integral part of our environment. Smaller chips, emerging wireless communication standards, and more capable sensors and actuators are pushing the development towards wireless sensor networks. Wireless sensor networks are composed of multifunctional miniature sensor devices with limited processing and storage capabilities, often battery operated, and interconnected with each other over wireless links. Over the past years, networking sensors have received more attention in various sectors. One application area is a health Body Area Network (BAN), consisting of a network of sensors carried by, and moving around with, the patient. The benefits from mobile (wireless) monitoring in out-of-hospital environments have been the focus in several studies and a number of prototypes have been developed e.g. [5-7]. These systems incorporate a range of wireless devices with varying capabilities. Even though a small number of nodes might be carried at the same time, the network of sensors may be configured quite differently over time (e.g. depending on how the medical condition of a patient progresses). Another application area is environmental monitoring where a number of different sensors can be used. Water level, temperature, and wind sensors can be deployed along a river bank, issuing

automatic warnings in the case of possible flooding [22], [23]. The common basis for the above application areas is the introduction of new sensors, often constructed by various manufacturers. This requires a specific understanding of the type of signal produced by the device and the proprietary protocol for communicating with (and controlling of) the sensors. Furthermore, deploying a large number of sensor nodes requires a high-level framework enabling the devices to present themselves and the services they offer. Using a service discovery protocol, it becomes possible to make sensor data available to an application through high-level interfaces.

In this paper, we review a number of well-known established service discovery architectures, which could be used to manage the complexity of sensor networks. Our target platform is a resource constrained sensor node, hence we seek a lightweight scheme that enables devices and their services to auto-configure, cooperate, adapt to changes, and to dynamically advertise and find available services in a sensor network. Bonjour [1], based on well-known standards, stands out with its auto-configuration, distributed ad hoc architecture, sharing of resources, and potentially lightweight implementation. To demonstrate that an emerging standards-based service discovery protocol actually can be deployed on small sensor nodes, we develop a lightweight Bonjour implementation. Initial experiments conducted on our prototyping sensor platform MULLE [12], verifies that service and device discovery of resource limited nodes in a sensor network is feasible.

This paper is structure as follows. In Section 2, common characteristics and a short survey of some of the well-known service discovery protocols are presented. Our development and prototyping platform is introduced in Section 3. In Section 4, our sample implementation of the Service Discovery Protocol is presented and in Section 5, future work is discussed.

## II. SERVICE DISCOVERY PROTOCOLS

Service discovery protocols enable services and service users to dynamically advertise and find available services in a network. They provide the necessary means to describe services so that the service users can determine if a discovered

---

This work was sponsored by the Centre for Distance-spanning Health-care (CDH) and by European structure funds in the project SIV ("Distance-spanning technology as support for cooperation within elderly home care").

service matches its requirements as well as utilize this service. Today, there exist a number of proposed service discovery protocols and the common building blocks and techniques of service discovery protocols include:

#### A. Service catalogues

Service discovery protocols can be categorized as either a centralized directory-based protocol or distributed directory-less protocol. In the former, nodes register their available services with a central repository where service users query for available services. In the latter scheme, the protocol is inherently peer-to-peer and the service catalogue is distributed over the sensor nodes.

#### B. Service description

The service discovery protocol must define a data description language, representing and describing the service. In addition, the additional capabilities of the service, or attributes, usually have a standard naming convention.

#### C. Registration & discovery

For service users (e.g. clients) to be able to find each other, services must be registered and a discovery process has to take place. To discover services, the discovery process can either be active (by issuing queries) or passive (by listening on service announcements from peer nodes).

#### D. Utilization

Other important characteristics for service discovery protocols are the techniques for supporting service delivery and service invocation. For some service discovery protocols, the responsibility for service invocation is controlled by higher level protocols apart from the actual service discovery protocol. Other protocols provide the necessary means to utilize the service by exporting a service interface.

#### E. Service status

To maintain a consistent state, it is necessary that the service discovery has a mechanism to notify the service users, e.g. ensuring that a clients' knowledge of an announced service is still valid. Either a client can receive a change of a service state by receiving asynchronous notification of a specific event, or by frequently polling the service.

#### F. Service discovery in sensor networks

Depending on the application area and usage scenario, the service discovery protocol has a number of requirements. In e.g. pervasive environments, it can be expected that the service discovery protocol must be able to cope with a number of different devices. In such environments, it is anticipated that devices are heterogeneous, ranging from very resource limited tiny nodes to more resource rich devices, e.g. PDAs and laptops carried by human users. For small devices, the processing power, storage capabilities in terms of memory, and communication capabilities must be taken into consideration. In addition, nodes are expected to be in a low-power state with a low duty cycle to conserve power. If the node is not reachable

in this state, cooperative techniques must be handled by the service discovery protocol.

#### G. Well known service discovery protocols

Well known service discovery protocols include Service Location Protocol (SLP), Jini, UPnP, and Bonjour. Table 1 summarizes our comparison of these protocols. The presented service discovery protocols have taken different approaches to enable dynamic service registration, discovery, and service invocation. For example, Bonjour and UPnP have a clear focus on enabling address allocation without DHCP servers, automatic discovery of computers, devices, and services on IP-based networks (as known as zero-configuration networking). Furthermore, in Jini, services are delivered as Java objects to service users requesting the service, making it possible to perform ordinary method calls. Service delivery and invocation in e.g. SLP and Bonjour, is on the other hand entirely left out from the protocol description.

#### H. Choice of service discovery protocol

The choice of service discovery protocol for our sensor node is based on a number of properties associated with ad hoc sensor networks. The storage, processing, and communication capabilities precludes some of the mentioned protocols. In such environments, initiatives have been made to support resource-constrained devices, for example the Jini Surrogate Architecture (e.g. for devices without a Java Virtual Machine) [15]. In this framework devices may join the service federation with the aid of a surrogate host, a resource rich device, representing and acting on behalf of the non-Jini capable device. On one hand, the Jini Surrogate Architecture may solve the issue of nodes having limited capability, but on the other hand, it enforces clients searching for services provided by the device to apply Java/Jini technology. Others have outsourced large and complex tasks to dedicated and more powerful nodes, e.g. allowing small nodes to become a part of an UPnP environment [5]. However, both approaches require a hosting environment that must be provided by a resource rich device. Furthermore, due to intermittent network connectivity in ad hoc environments, nodes may appear and disappear without notification. For an application that needs to maintain a consistent view of the available services, the node may either poll the network repeatedly or receive a notification when a change occurs. As seen in Table 1 Jini, UPnP, and Bonjour support the notion of detecting a change of the service status state by either polling or receiving a notification event. SLP relies on polling, but work has been performed to support notification as well to detect changes [21]. In addition, in ad hoc networks with devices acting as routers and hosts at the same time, forming an arbitrary topology, it is necessary that the service discovery protocol does not rely on a centralized architecture (e.g. as Jini presuming the existence of a central repository for service registration and service lookup). Finally, dynamic service discovery protocols are often designed to be scalable in local networks [6], [13]. Extending from searching and browsing for services in radio proximity, the possibility to register services and perform lookups from the global Internet would be beneficial. SLP and Bonjour present solutions to operate in a local scope as well as searching for services in a global scope. This is achieved by utilizing extensions to

TABLE 1 SHORT OVERVIEW OF SERVICE DISCOVERY PROTOCOLS

	SLP [6]	Jini [16]	UPnP [17]	Bonjour [1]
<i>Service catalogues</i>	Centralized or distributed	Centralized	Distributed	Distributed
<i>Capability description</i>	Service templates	Interface and Entry objects	XML device templates	DNS TXT records
<i>Service registration</i>	Unicast to DA or multicast advertisements	Contact lookup service	Multicast advertisements	Multicast advertisements
<i>Service discovery</i>	Unicast to DA or multicast to SA	Query to lookup service	Multicast query	Multicast query
<i>Utilization</i>	Unspecified	Proxy objects	SOAP	Unspecified
<i>Service status</i>	Polling only	Polling or notification	Polling or notification	Polling or notification

existing standards (DNS) thus enabling remote service discovery. Since Bonjour in particular is based on DNS, whereas SLP adds it for remote service discovery, we have selected the Bonjour device and service discovery protocol as being suitable for our implementation on the sensor nodes.

### III. NODE PLATFORM OVERVIEW

In the past, a number of prototypes of wireless networking sensor nodes have been developed. Many of these devices are built using commercial-off-the-shelf (COTS) components. COTS hardware platforms such as the Berkley Mica motes [10] have often been used when developing applications. Another platform is the BNode [3], a demonstration and research platform. Our representation of a node (Fig. 1) is similar. The major hardware components are a 16-bit single-chip microcontroller, a Bluetooth single-chip module with integrated antenna, and an interface to connect sensors and actuators. The software architecture consists of lightweight versions of TCP/IP and Bluetooth stacks. Utilizing standard protocols for communication has several advantages over developing proprietary protocols. For example, it makes it possible for nodes to operate seamlessly with different types of devices (other nodes, access points, mobile phones, PDAs etc). Also, Bluetooth is a widely accepted wireless standard and enables short-range wireless data communication between devices. It has been argued that a wireless sensor node, having the limited computation, memory, and communication resources is precluded from the use of the “heavyweight”

networking protocols [9], [14]. It is true that the components assembled on the development platform incorporate more powerful components than normally found on small sensor nodes [18], [11]. However, it has been shown that the TCP/IP protocol can be used on COTS devices, similar as those traditionally used as sensor nodes [4]. Finally, achieving interoperability with a large number of devices makes Bluetooth valuable, especially when prototyping sensor-based applications.

### IV. DESIGN CONSIDERATIONS

Our lightweight implementation of Bonjour enables the sensor node to announce its services in a small network of devices in the dot-local domain. This presentation of services requires the capability of announcing and responding with appropriate DNS messages to peer nodes. A significant portion of the DNS messages is used by the representation of domain names. On our platform, the scarcest resource on the node is memory. The 16-bit microcontroller on the platform has 256 kB of flash memory for program code, and 20 kB RAM. Given that in the target environment, almost all information is known at compile time, i.e. the service or set of services, hence we store the appropriate domain names in ROM reducing the amount of dynamic memory needed to be allocated during runtime. Information not known at compile time, which has to be allocated during boot, is typically the node host name, IP address and the instance part of the Bonjour service instance name. The intention of the instance part is to represent a user-friendly name, containing any UTF-8-encoded text. Table 2 shows the size of the code compiled with [19] and executed on the M16c 16-bit microcontroller. In this example, one service is announced by the node. As an indication of the size, we also compiled the code from the Bonjour project [20] for the M16c platform and the 32-bit Intel x86- architecture. As an evaluation, those figures cannot be directly compared; the target environments are simply different. For example, the open source code supports much larger DNS messages (up to Ethernet Jumbo frames). However, in our target network technology, such large DNS messages are unlikely to be generated and as a result, we have opted to support the original maximum size of UDP DNS messages (512 bytes). This also gives us an estimation of the memory consumption when

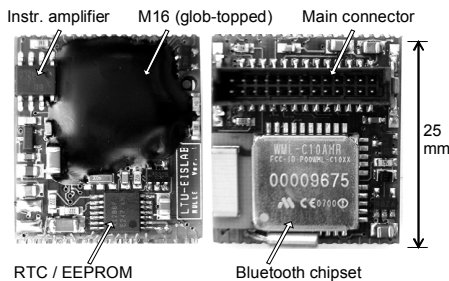


Figure 1 Our prototype platform. Both sides of a board with the main components indicated.

TABLE 2 CODE SIZE AND RUNTIME ALLOCATION OF MEMORY WHEN ANNOUNCING ONE SERVICE.

	Lightweight code	Bonjour code
Code & constant size	11 kB	132 kB (M16c architecture)
RAM usage, 1 service	329 bytes	11 kB (x86 architecture)

processing e.g. a query. Even though multiple records might exist in a DNS query message, they are processed one at a time marking potential answers of the nodes services. Assuming the maximum sized DNS message, and the maximum resource record extracted and supported by the node, the processing of queries needs to allocate at most 1 kB of memory during runtime.

#### A. Sample implementation

When started, an onboard control application initiates an inquiry to search for other Bluetooth devices in the close proximity. The Bluetooth standard defines a set of profiles for communication. Our platform currently supports the LAN Access Profile (LAP) and the Dial-Up Network (DUN) profile to access a remote network. The peer device may be a mobile phone with GPRS or a Bluetooth access point, connected to a wired network. When connected, a node joins the multicast address assigned for Bonjour, and announces its services (currently, we experiment with announcing two services). The first service is a web-server, announced using the appropriate DNS PTR record with the name `_http._tcp.local.`, and with record data pointing to the specific sensor node. The scenario is depicted in Fig. 2, where the Internet Explorer plug-in is used to show the HTTP services found in the local domain. In this example, a standard PC is connected to a wired Ethernet network as well as connected over Bluetooth acting as a Data Terminal (LAP-DT). The plug-in multicast DNS queries for e.g. PTR records to browse the dot-local domain to find out all available web servers, and receives a number of responses. The HTTP service is conveniently utilized, as shown in the figure,

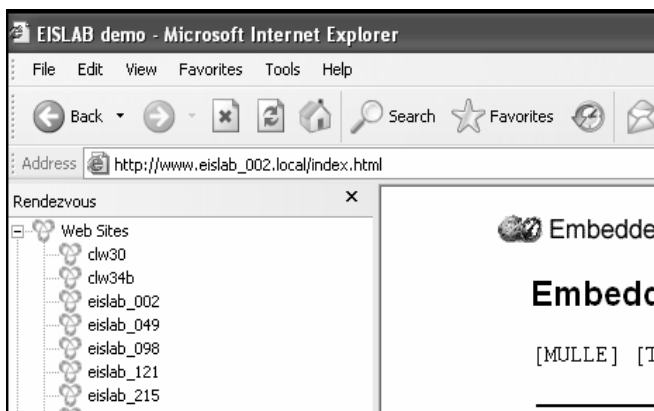


Figure 2 Snapshot of Internet Explorer with Bonjour plug-[2] in when connected in a Bluetooth piconet.

by clicking on the appropriate instance found by the lookup process.

The second service is the application-specific service, depending on type of sensor attached to the platform. Currently, the abstract service type 'eis' is used, matching any type of physical sensor connected to the development platform. This usage relies on a higher level application protocol in order to utilize the service i.e. the client application must know the specific details how to access data sampled by the sensor, as well as the format and type of the data. In the case of people accessing the platform by the use of a standard browser, the application protocol could be embedded in a Java applet downloaded from the web server. The use of a generic service type to represent the physical sensor demonstrates one of the strengths with Bonjour. In fact, the service discovery protocol supports any application level protocol running on IP based networks.

#### V. FUTURE WORK

Radio communication is the most significant cause of energy consumption for our prototyping sensor platform, and device energy consumption is a primary issue in mobile applications. The Bluetooth specification defines piconets (groups of up to 255 device that are directly addressable) where eight of these devices can be active at any given time. In the future, we intend to investigate a Sleeping Proxy service type in a Bluetooth piconet. Implemented on a master node, the node will be able to answer general queries on the behalf of another device. This would allow us to put the slave sensor node in sleeping mode, thus reducing energy consumption.

#### VI. CONCLUSION

This paper presents the feasibility of the deployment of a lightweight service discovery protocol on wireless sensor networks. The target platform is a small sensor node which implements an ad hoc sensor networking device with IP and Bluetooth protocol stacks. We have shown that the nodes are capable of advertising and hosting services in the .local domain in a Bluetooth piconet. The lightweight implementation of the emerging device and service discovery protocol is based on well known standards-based communication protocols, thus providing a familiar environment when developing applications for the sensor network. Our initial experiments using the platform verify that service and device discovery of nodes in an ad hoc sensor network is feasible.

#### REFERENCES

- [1] Apple Computer, Inc., Rendezvous Technology Brief. March 2005. [http://www.apple.com/macosx/features/rendezvous/Panther\\_Rendezvous\\_TB\\_10232003.pdf](http://www.apple.com/macosx/features/rendezvous/Panther_Rendezvous_TB_10232003.pdf).
- [2] Apple Computer, Inc., Rendezvous for Windows Technology Preview 4. <http://developer.apple.com/macosx/rendezvous/index.html>, Last visited: Feb 2005.
- [3] Beutel, J., and Kasten, O., and Mattern, F., and Roemer, K., and Siegemund, F., and Thiele, L., Prototyping Wireless Sensor Network Applications with BTnodes. In 1st IEEE European Workshop on Wireless Sensor Networks (EWSN), Berlin, 19-21 Jan 2004.

- [4] Dunkels, A., Full TCP/IP for 8-Bit Architectures. In Proceedings of the first international conference on mobile applications, systems and services (MOBISYS 2003), San Francisco, May 2003.
- [5] Gsottberger, Y., Shi, X., Stromberg, G., Sturm, T.F., and Weber, W., Embedding low-cost wireless sensors into universal plug and play environments. In *Wireless Sensor Networks. First European Workshop, EWSN 2004. Proceedings. (Lecture Notes in Comput. Sci. Vol.2920)*, pp. 291 - 306, 2004.
- [6] Guttman, E., Perkins, C., Veizades, J., and Day, M., Service Location Protocol, Version 2, RFC 2608. June 1999. IETF.
- [7] Halteren A.T. van, Bults R., Widya I.A., Jones V., and Konstantas D., Mobihealth-Wireless body area networks for healthcare, in "Wearable eHealth Systems for Personalised Health Management", Vol. 108 "Studies in Health Technology and Informatics", A. LyMBERIS and D. de Rossi (Ed.), Il Ciocco Castelvechio Pascoli Lucca, Tuscany, December 11-14th 2003, ISBN 1 58603 449 9, IOS press, Amsterdam, 2004;
- [8] Korhonen, I., Parkka, J., and Gils, M. Van, Health monitoring in the home of the future. In *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, pp. 66-73, May-June 2003.
- [9] Malan, D., Fulford-Jones, T., Welsh, M., and Moulton, S., CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. In *International Workshop on Wearable and Implantable Body Sensor Networks*, April 2004.
- [10] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K., System Architecture Directions For Networked Sensors. In Proceedings of the ninth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 93-104, 2000.
- [11] Hill, J., Horton, M., Kling, R., and Krishnamurthy, Lakshman, The platforms enabling wireless sensor networks. In *Commun. ACM, ACM Press*, vol. 47, pp. 41-46, 2004.
- [12] Johansson J., Völker V., Eliasson J., Östmark Å., Lindgren P., Delsing J., MULLE: A Minimal Sensor Networking Device – Implementation and Manufacturing Challenges. In *Proceedings of the IMAPS Nordic Annual Conference*, pp. 265-271, September 2004.
- [13] Lee, C., Helal, A., Verma, N., Desai V. and Arslan, B., Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. In *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 33, pp. 682-696, Nov 2003.
- [14] Marcy, H.O., Agre, J.R., Chien, C., Clare, L.P., Romanov, N., and Allen, T., Wireless sensor networks for area monitoring and integrated vehicle health management applications. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR*, Collection of Technical Papers. (A99-36576 09-63), (AIAA Accession number 42108), Vol. 1 Aug. 9-11, 1999.
- [15] Sun Microsystems, Jini™ Technology Surrogate Architecture Specification, v1.0 Standard.Oct 2003. <http://surrogate.jini.org/sa.pdf>.
- [16] Sun Microsystems, Jini™ Architecture Specification, Version 2.0. June 2003. [http://www.sun.com/software/jini/specs/jini2\\_0.pdf](http://www.sun.com/software/jini/specs/jini2_0.pdf).
- [17] UPnP Forum, UPnP™ Device Architecture v1.0.1 Draft. Dec 2003. [http://www.upnp.org/download/UPnPDA10\\_20000613.htm](http://www.upnp.org/download/UPnPDA10_20000613.htm)
- [18] M.A.M. Vieira and C.N. Coelho Jr. and D.C. da Silva Jr. and J.M. da Mata., Survey on wireless sensor network devices. In *Proceedings IEEE Conference Emerging Technologies and Factory Automation*, vol. 1, pp. 537-544, 19-19 Sept 2003.
- [19] IAR Systems. IAR C/EC++ Compiler for M16C V2.12A/W32 (2.12.1.4)
- [20] Bonjour Developer Web Site, <http://developer.apple.com/darwin/projects/bonjour/>, Last visited: June 2005.
- [21] J. Kempf and J. Goldschmidt, Notification and Subscription for SLP, 3082 March 2001. IETF
- [22] S. Tilak, N.I B. Abu-Ghazaleh and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," in *SIGMOBILE Mob. Comput. Commun. Rev.*, ACM Press, vol. 6, pp. 28-36, 2002.
- [23] M. Tubaishat and S. Madria, "Sensor networks: an overview," in *IEEE Potentials*, vol. 22, pp. 20 - 23, April-May 2003.