A Comparison of Approaches to Node and Service Discovery in 6lowPAN Wireless Sensor Networks

Ricardo Silva
Departamento de Engenharia Informatica, Universidade de Coimbra
Polo II - Pinhal de Marrocos, 3030-290
Coimbra, PORTUGAL
rnsilva@dei.uc.pt

V. R. Q. Leithardt
Sistemas Largamente Distribuidos, Universidade Federal do Rio Grande do Sul, Instituto de Informatica
Av. Bento Goncalves, 9500-Campus do Vale, Bairro Agronomia
Porto Alegre - RS - BRASIL
vrqleithardt@inf.ufrgs.br

Jorge Sa Silva
Departamento de Engenharia Informatica, Universidade de Coimbra
Polo II - Pinhal de Marrocos, 3030-290
Coimbra, PORTUGAL
sasilva@dei.uc.pt

Claudio Geyer
Sistemas Largamente Distribuidos, Universidade Federal do Rio Grande do Sul, Instituto de Informatica
Av. Bento Goncalves, 9500-Campus do Vale, Bairro Agronomia
Porto Alegre - RS - BRASIL
geyer@inf.ufrgs.br

Joel Rodrigues
Instituto de Telecomunica-coes,Department of Informatics, University of Beira Interior
Rua Marques D'Avila e Bolama, 6201-001
Covilha, PORTUGAL
joel@ubi.pt

Fernando Boavida
Departamento de Engenharia Informatica, Universidade de Coimbra
Polo II - Pinhal de Marrocos, 3030-290
Coimbra, PORTUGAL
boavida@dei.uc.pt

ABSTRACT

Discovering nodes and services in Wireless Sensor Networks poses several challenges. Different sink and sensor nodes announcement strategies lead to different amounts of resource consumption in terms of processing, memory, communication time and drained energy. This may be aggravated if IP is the underlying protocol, such as in the case of 6lowPAN networks. In this paper three paradigms for node and service discovery are proposed, analyzed and evaluated. The results, obtained by prototyping, show that at least one of the proposed strategies leads to efficient use of resources, showing that node and service discovery in resource-constrained networks such as 6lowPAN is feasible.

Categories and Subject Descriptors

C.2 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Service and Node Discovery, Wireless Sensor Networks

1. INTRODUCTION

Recent developments in sensor networks have been driven by the combined effect of enabling technologies and pressing application needs. The basic components of sensors nodes, also called motes, are the transducers, elements capable of measuring physical parameters, such as temperatures, light intensity or pressure, and of converting them from analog to digital form. Another fundamental component is the communication module, supporting wired or wireless communication. The combination of wireless communication capabilities with sensing capabilities and data processing capabilities has led to the concept of wireless sensor network (WSN) and to a myriad of potential applications. However, many of these applications require that the sensor nodes are deployed in places that are not easily reachable or that are even unknown. Thus, motes are basically standalone devices that, after deployment, are not recovered and must operate with their initial energy source only. Under these conditions, processing, memory and communications restrictions are fundamental in order to extend the life of the mote and, consequently, of the WSN, as far as possible. The TinyOS [1] sensor node operating system provides an efficient use of the nodes' hardware capabilities and also allows the installation of specific sensor-oriented programs written in the nesC programming language [2]. In addition to operating systems, motes need communication protocols to convey the data they gather.

The architecture of a WSN is based on the deployment of simple motes that communicate with a central node called the sink node. The sink node receives the measured data from motes and processes it locally or sends it to a central unit. Normally, this communication is supported on some kind of Layer 2 protocol. Currently the most common Layer 2 communication protocol used in WSNs is the IEEE 802.15.4 protocol [3].

Using Layer 2 protocols to communicate is the most common approach, since it is quite inexpensive from an energy point of view and most communication is point-to-point (from node to sink). However, it has several important limitations, the major one being the lack of global IP connectivity. The idea of accessing WSNs directly from the Internet greatly enhances the potential and applicability of wireless sensor networks. Based on this idea, a new IETF group, the 6lowPAN group [4], proposed the use of IP in motes. Specifically, the 6lowPAN working group proposed a special layer to provide IPv6 [5] connectivity over the IEEE 802.15.4 Layer 2 protocol, having in mind the power and cost restrictions that characterize sensor nodes. The 6lowPAN work items are [6], [7]:

- IP adaptation / Packet formats and interoperability
- Addressing schemes and address management
- Network management
- Implementation considerations
- Routing in dynamically adaptive topologies
- Security, including set-up and maintenance
- Application programming interfaces
- Discovery (of devices, services, etc).

As result of their work, the 6lowPAN working group launched two RFCs, related to the first two work items: RFC 4919 [6], which provides an overview, assumptions, problems and goals for improving IPv6 over low power wireless PAN; and RFC 4944[7], which specifies the transmission of IPV6 Packets over IEEE.

Most of the existing WSN communication protocols do not implement mobility support and were designed for a totally stationary scenario. However, some MAC layer protocols already provide some support for node mobility as well as for services discovery. MS-MAC [8], MAMAC [9] and MH-MAC [10] are just examples.

This paper presents and proposes three approaches to device and service discovery in wireless sensor networks. The first approach, called YouCatchMe, follows the philosophy of conventional wireless networks, where access points periodically send beacons, announcing their presence. The second approach looks at the issue from a different perspective: in the ICatchYou approach, sink nodes are silent and it is the responsibility of each sensor node to search and detect sink nodes. Lastly, Some1CatchMe makes use of RFID tags and readers to discover sensor nodes. All of the proposed approaches were implemented, in order to study and evaluate them.

This paper is organized as follows. Section II introduces and describes the proposed paradigms for service and node discovery. Section III provides a comparative analysis of the presented approaches, followed by Section IV, in which this comparison is extended to the results obtained through implementation, in order to better assess the impact of the proposed service discovery mechanisms, as well as the impact of using IP-based solutions. Conclusions and guidelines for further work are presented in Section V.

2. APPROACHES TO NODE AND SERVICE DISCOVERY IN WIRELESS SENSOR NETWORKS

2.1 Motivations for node and service discovery

Although WSNs can be used in static environments, dynamic environments are the ones for which WSNs have the highest potential use. In fact, if we consider typical WSN characteristics, such as energy constraints, unmanned deployment and operation, almost all designed WSNs are theoretically dynamic, even if the nodes are fixed. Dynamic environments typically are designed and account for node mobility. In such an environment, a node should be able to dynamically switch between different WSNs and perform auto-configuration, repeating the process whenever there is the need for it, several times during the mote's lifetime. As example scenario, we can consider a hospital where each patient carries one or more sensors that are monitoring various life signals. Some patients are bedfast, while others can walk around the facilities, for instance, to go into the main room to watch television, or to have lunch at the canteen. In order to get the sensor nodes measured values, the hospital can install one sink node per room. Each time patients move from one room to another, their own sensor nodes have to be integrated into the new WSN of that room. As this example shows, a WSN discovery system is required, not only for devices but also for services, in order to enhance and take advantage of automatic monitoring environments.

2.2 Overview of proposed approaches

Three approaches to node and service discovery are proposed in this paper: YouCatchMe, ICatchYou and Some1CatchMe. This sub-section provides an overview of each of these approaches.

The YouCatchMe node/service discovery paradigm relies on active sink nodes: a sink node periodically broadcasts announcements (Router Advertisements). Upon arrival at a new network, wireless sensor nodes receive the sink node announcements, initiate the registration procedure and indicate the services they support to the sink node. The sink node knows in real time which nodes are in the area and what services they provide.

The ICatchYou node/service discovery paradigm is based on a completely different approach: instead of having the sink node sending announcements, it is the sensor node that, after deployment or when it arrives at a new network, starts sending requests for registration. This eliminates the need for sink node broadcast announcements (as opposed to YouCatchMe) but introduces a sensor node announcement. This also leads to a simpler and more efficient discovery protocol, when compared to the IPv6 Neighbor Discovery procedure. When a sink node receives a registration request it initiates the procedure to register the new sensor node and to get information on the supported services, using a unicast Router Advertisement (RA).

The Some1CatchMe discovery paradigm is based on a multi-radio approach, as it resorts to RFID and IEEE802.15.4. In order to avoid the periodic broadcasts of YouCatchMe or the Router Solicitations of ICatchYou, Some1CatchMe uses RFID to discover the new nodes within the network range. Upon arrival at a new network, nodes equipped with an RFID tag are discovered by a local RFID reader, which is connected to the local sink node. After detection of a new sensor node, the sink node initiates the registration procedure, communicating directly with this new node. Since all power used in RFID comes from the RFID reader, motes do not spend energy during the discovery procedure.

RFID technology has been used in several applications, especially those that require low power consumption. There are four types of tags: low frequency tags - 126 to 134 kHz (1m), high frequency tags - 13to56 Mhz (15m), UHF tags - 868 to 956 MHz (> 15m), microwave tags - 2.45 Ghz (> 1.5 m) [11].

In the next sub-sections each of these approaches will be presented in more detail, namely in what concerns the discovery and registration phases and message exchanges.

2.3 C. YouCatchMe

In YouCatchMe, sink nodes send beacons to announce their presence. These use the ICMPv6 protocol [12] in order to periodically send Router Advertisements (RA) to the broadcast address. After deployment or upon arrival at a new WSN, sensor nodes start receiving the beacons and answer with an acknowledgement. On their turn, sink nodes validate the received acknowledgement messages and maintain a database of registered sensor nodes. This database can be used by applications to know which sensor nodes, and respective services, are available. The YouCatchMe operation is based on a fixed cycle comprising three different periods: the B (broadcast) period, in which RA messages are broadcasted; the L (listening) period, in which acknowledgement messages are received and processed; and the S (sleep) period, during which no node discovery activity takes place, leaving room to data exchange with the sensors. These are represented in Figure 1.

Figure 1: YouCatchMe operation time line.

The choice of the cycle duration has direct consequences on the number of broadcast messages and thus, on energy consumption, as these messages are processed by all nodes. If a node does not respond to a beacon, the sink assumes that it has moved out or died. Sink nodes maintain a database containing only the nodes that responded to the most recent beacon, and these are the ones that are considered active in the current period Pi, where $i \in N$. For instance the nodes detected in $P1$ could be different from the ones detected in $P2$.

2.4 ICatchYou

ICatchYou was designed to minimise the exchange of broadcast messages. In this approach the deployed nodes have the responsibility of finding the most suitable sink node and start the registration process. The ICatchYou operation comprises the following:

- Gathering information about all sink nodes in the area and choosing the one with the best link quality.;
- Self-configuration of an IPv6 Global Address, using as prefix the one of the chosen network and as suffix the one of the Interface Identifier. The Interface Identifier could be the 64 EUI MAC address or generated from the 16 bits short address, as defined in [7].
- Use of a TTL value defined by the sink node. This TTL represents the validity of the current registration.
- Capacity to detect that the TTL value is almost expired and, in this case, to initiate the update of the current registration.

The underlying discovery/registration protocol comprises an initial registration phase, a TTL phase and subsequent update/TTL phases (see Figure 2).

Figure 2: ICatchYou operation time line.

The proposed protocol was designed and implemented as an ICMPv6 extension, using free messages types for each protocol step. It is clearly an extension of Neighbor Discovery, using also the Router Solicitation (RS) and Router Advertisement (RA) messages

Figure 3 shows the implemented registration and update procedures.

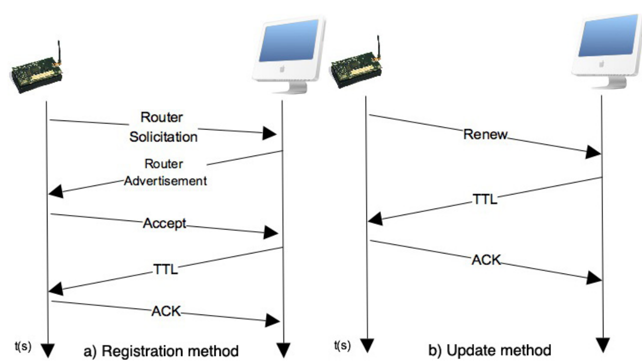

Figure 3: ICatchYou protocol.

The registration procedure (Figure 3a) starts with a Router Solicitation (RS) message, sent from a node to the sink nodes multicast/anycast address. Upon reception of an RS message, sink nodes reply with an RA message. Through this reply the node checks the link quality, selects the sink node with the best link quality and sends it the ACCEPT message. After reception of an ACCEPT message, the sink node assigns a TTL value to the new node. When the node receives the TTL, it self-configures its IPv6 global address, according to the network prefix of the sink node, replies with an ACK message, initiate a timer with the TTL value and proceeds to countdown until achieves the update phase (U) threshold value.

When the node enters the update phase, it sends the RENEW message directly to its sink node. In normal conditions, the sink node receives the command, checks the database to validate it and assigns a new TTL value. Otherwise, if the sink node is no longer available, the update period will expire and the node will start the registration process.

A study of TTL impact is presented in section IV, below.

2.5 Some1CatchMe

Some1CatchMe makes use of RFID tags associated to IEEE 802.15.4. RFID tags do not process anything. They just contain an identification that can be read by RFID readers. All energy to read the tag comes from the RFID reader; hence, if we associate a tag to a node, the RFID reader can discover it without energy consumption by the node.

Thus, upon arrival of a sensor node to a new network, the sink node RFID reader reads the RFID tag of the sensor node, containing the IEEE 802.15.4 EUI 64 bits address. This allows the sink node to communicate with the new sensor node, using the Link Local IPv6 address obtained from the Layer-2 64-bits address.

When a sink node detects a new node, it initiates a registration process similar to the one presented for ICatchYou, using a TTL as well. Comparing the registration procedure of Some1CatchMe with that of ICatchYou (Figure 3a), the Router Solicitations disappear. Thus, the node discovery process is completely transparent to the sensor node itself. The node only becomes aware of the registration process when it receives a unicast RA from the sink node. Considering that the node can receive RAs from different sink nodes, it can also choose the best one based on the link quality, as in the ICatchYou case.

The Update method is the same as in ICatchYou.

3. COMPARATIVE ANALYSIS

3.1 YouCatchMe versus ICatchYou

The YouCatchMe paradigm is inspired in the most common way of operation of wireless networks: access points announce their presence through a beacon signal. However, when implemented over a WSN, this paradigm leads to inefficient use of resources, mainly caused by the impossibility of avoiding that registered nodes receive the announcements (beacons), which, in turn, leads to waste of energy. In fact, as sensor nodes use wireless technology, an intrinsically shared environment, it is impossible to avoid the reception of broadcast messages by all nodes in range. In [13] this limitation was overcome by the suppression of Router Advertisements.

The excessive number of broadcast messages always limits YouCatchMe, even when it is carefully configured. In the YouCatchMe approach the number of broadcast messages during a given time interval is dictated by the duration of this interval, P, and the period between broadcasts, T:

$$B = \frac{P}{T} \tag{1}$$

The number of broadcasts in the same period for the ICatchYou approach depends on the number of RSs sent, which is dependent on the total number of nodes that requested a registration during P.

$$B = \sum_{i=1}^{n} X_i \tag{2}$$

In (2) n is the number of nodes that requested registration during $P.B$ is the number of broadcasts sent by node i during the period P. In (1) B is inversely proportional to T. In (2) B increases with network dynamism and with the degradation of link quality. Under optimal conditions, each node only sends one broadcast during P, and thus, in this case:

$$B = n \tag{3}$$

Therefore, with a network constituted only by highly dynamic nodes, which are constantly switching between networks, the value yielded by (2) can be greater than (1), in the same time frame P. However, even in this worst scenario, it continues to be an adaptable variable, which always fits the network conditions under different scenarios. In (1), the number of broadcast can be smaller than the one in (3), but in this case this means that the network is under highly conditions and, most likely, the nodes would not be detected and would consequently be unusable. Hence, ICatchYou is a sink discovery and registration method that optimally adapts to the network and node conditions, leading to better response and availability.

3.2 ICatchYou versus Some1CatchMe

The main objective of Some1CatchMe is to reduce energy consumption by simplifying the discovery and registration procedures. Passive RFID tags operate without batteries [26] as the reader provides the necessary energy through the electromagnetic waves. RF passive tags are cheaper than active tags and their lifetime is theoretically unlimited. RF passive tags are usually read-only, are used for short distances and require a more complex reader.

Theoretically, from an energy point of view, Some1CatchMe is more efficient than ICatchYou. Some1CatchMe uses RFID to discover new and/or mobile nodes. Hence, no broadcast messages are sent, avoiding the Router Solicitation step of the ICatchYou approach.

However, the integration of RFID readers with sink nodes has some costs. Moreover, the necessary conditions for installing and using Some1CatchMe are largely application-dependent, making this approach less general. Scalability issues may also be relevant.

4. IMPACT OF NODE/SERVICE DISCOVERY MECHANISMS

In order to evaluate the impact of the presented architectures, from an energy consumption point of view, a real test was performed. All studies were developed over a 6lowPAN implementation [14], included in the latest TinyOS CVS version. This implementation supports UDP communication and ICMPv6. The used hardware was the MicaZ motes and the mib520 USB board, both from Crossbow [15].

The three mechanisms - that is, YouCatchMe, ICatchYou and Some1CatchMe - were implemented. Subsequently, energy consumption was calculated, based on the measurement of the electric current (mA). The electric current of motes highly depends on the operation mode, more specifically, on the fact that the mote radio transceiver is on or off, and on if it is transmitting or not. The amount of energy spent in a given time frame, t, can be calculated using equation (4), where V is the battery voltage (approximately constant) and I is the measured current. The total amount of energy consumed in a mixed operation period (i.e., with radio off, radio on, and radio on with transmission) is given by equation (5).

$$E_t = V \times I \times t \tag{4}$$

$$E_{Total} = E_{radioOff} + E_{radioOn} + E_{radioTransmitting} \tag{5}$$

In all evaluations were used a static energy source with 2.7Volts and the mote transceiver was always on. In YouCatchMe, for each received RA motes send an ACK. These two messages (RA and ACK), took an average time of 49.3ms, in the performed experiments.

Table 1 presents the values obtained for the YouCatchMe energy consumption during one hour of test. $E(mJoules)$ directly derives from formula (4), with voltage equal to 2.7Volts. T is the period between broadcasts, in seconds.

Table 1: YouCatchMe energy consumption

	Radio On w/o transmitting	Radio transmitting
Measured electric current (mA)	22.5	23.4
$t(s)$	$3600 - (\frac{3600}{T} \times 0.0493)$	$\frac{3600}{T} \times 0.0493$
$E(mJoules)$	$2.7 \times 22.5 \times t$	$2.7 \times 23.4 \times t$

In order to determine the energy required by ICatchYou, the mote already used before for YouCatchMe was programmed with the ICatchYou mechanism. Considering the same operation modes, ICatchYou had an initial registration procedure manually configured to 5 seconds and various, subsequent update procedures, depending on the assigned TTL. So, the energy spent while transmitting was given by equation (6):

$$E_{RadioTransmitting} = E_{Register} + E_{Update} \tag{6}$$

In this experiment, each update took an average of 48ms. The ICatchYou energy consumption is presented in Table 2.

Since Some1CatchMe uses RFID to perform node discovery, it only uses the radio transceiver to update the registration. Hence, the energy consumption is the same as in ICatchYou without the registration procedure, as presented in Table 3, below.

Based on the values presented in Tables 1 to 3, Figure 4 shows the additional energy consumption in one hour, for the three mechanisms, for $T = TTL =$ [10,20,30,40,50,60, 70,80,90,100] seconds. The additional energy consumption is given by:

$$E_{add} = (E_{Radio_{on_{w/o}transmitting}} + E_{Radio_{t}ransmitting}) - E_{3600} \tag{7}$$

Table 2: ICatchYou energy consumption

	Radio On w/o transmitting	Radio transmitting	
		Reg.	Upd.
Measured electric current (mA)	22.4	22.8	22.7
$t(s)$	$3595 - (\frac{\frac{3600}{TTL}}{\frac{1}{0.048}})$	5	$\frac{3600}{TTL} \times 0.048$
$E(mJ)$	$2.7 \times 22.4t$	$2.7 \times 22.8t$	$2.7 \times 22.7t$

Table 3: Some1CatchMe energy consumption

	Radio On w/o transmitting	Radio transmitting
Measured electric current (mA)	22.4	22.7
$t(s)$	$3600 - (\frac{3600}{TTL} \times 0.048)$	$\frac{3600}{TTL} \times 0.048$
$E(mJoules)$	$2.7 \times 22.4 \times t$	$2.7 \times 22.7 \times t$

E_{3600} is the energy that the mote would spend only for maintaining the radio transceiver on, during 3600 seconds, without transmitting (i.e., only by listening to the environment).

As expected, Some1CatchMe requires less energy than the other two approaches, due to the fact that registration does not require message exchanges. YouCatchMe and ICatchYou have a particular behavior that, at first, gives advantage to ICatchYou for $T = TLL < 50$ and, for $n = TTL > 50$, to YouCatchMe. Of course, this is only an advantage from a theoretical point of view as, in practice, beacon periods of more than 50 seconds are too high and are never used. Although, for instance IEEE802.11 is defined to send a beacon at each 10ms, in WSNs broadcast should be avoided and such value is impossible to be applied. From an application point of view n should be as shorter as possible, however from an energetic prism it should be the longest possible, as proved in figure 4. Therefore, YouCatchMe is not the most suitable solution to be used in WSNs.

ICatchYou shows reasonable energy consumption for a reasonable mechanism, however great part of the energy required comes from the registration procedure, which can be improved or even removed, resulting in Some1CatchMe. The complexity of Some1CatchMe is relied on a multi-radio platform using RFID. Nodes become dynamically available in a new network, without any energetic effort. To choose between ICatchYou and Some1CatchMe a cost-effective relation must be taken in consideration, where the additional cost of the RFID system counterbalance or not the additional energy spent with ICatchYou.

5. CONCLUSION

In the current paper we have addressed the problem of

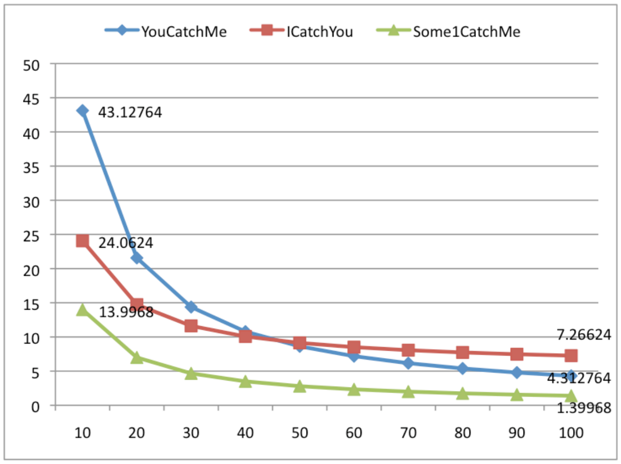

Figure 4: The impact of each mechanism in the energy consumption of the mote in one hour.

service and node discovery in wireless sensor networks using the 6lowPAN framework. In this context, three paradigms were proposed, analyzed and evaluated, having in mind the resource restrictions that are typical of WSNs.

Although the proposed solutions use IPv6 as the underlying protocol and require some simple higher layer mechanisms, the evaluation results clearly show that at least two of them - the ICatchYou and Some1CatchMe approaches - lead to good performance and thus not imply high resource consumption. Moreover, if the type of application allows it, Some1CatchMe is able to eliminate broadcasts, which makes it the most energy efficient method for node and service discovery.

Future work will be directed towards the improvement and refinement of the ICatchYou and Some1CatchMe approaches, specifically in what concerns the motes' capability to select the best access point, to self configure IPv6 global addresses, and to support single or multiple access points per network, among other. This will be fundamental for the success of the wider multi-sink, IPv6-based, mobility-enabled WSN architecture underlying the presented node and service discovery mechanisms.

6. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ną 224282 and by the Euro-NF Network of Excellence of Seven Framework Program of EU - IT Portugal.

7. REFERENCES

[1] The tinyos website http://www.tinyos.org/, 2009.

[2] The nesc website http://nescc.sourceforge.net/, 2009.

[3] Ieee std. 802.15.4 http://www.ieee.org/, 2009.

[4] Carsten Bormann and Geoffrey Mulligan. Ipv6 over low power wpan (6lowpan), May 2009. `http://www.ietf.org/html.charters/6lowpan-charter.html`.

[5] R. Hinden S. Deering, Nokia and Nokia. Internet protocol,version 6. RFC 2460, December 1998.

[6] N. Kushalnagar, G. Montenegro, and C. Shumacher. Ipv6 over low-power wireless personal ÃĄrea networks (6lowpans). RFC 4919, August 2004.

[7] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee802.15.4 networks. RFC 4944, September 2007.

[8] Huan Pham and Sanjay Jha. Addressing mobility in wireless sensor media access protocol. *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004*, pages 113–118, Dec. 2004.

[9] Liu Bing, Yu Ke, Zhang Lin, and Zhang Huimin. Mac performance and improvement in mobile wireless sensor networks. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, 3:109–114, 30 2007-Aug. 1 2007.

[10] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto. A wireless sensor mac protocol for bursty data traffic. *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, Sept. 2007.

[11] Rfid website www.epcglobalinc.org, 2009.

[12] Z. Shelby, P. Thubert, J. Hui, S. Chakrabarti, and E. Nordmark. Neighbor discovery for 6lowpan draft-ietf-6lowpan-nd-00, 2008.

[13] L. Toutain, G. Chelius, Y. Lee, and Y. Dong. Neighbor discovery suppression draft-toutain-6lowpan-ra-suppression-00.txt, 2008.

[14] M. Harvan. Connecting wireless sensor networks to the internet - a 6lowpan implementation for tinyos 2.0. In *Jacobs University Bremen, Germany*, 2007.

[15] The xbow website http://www.xbow.com, 2009.