A proposal for System Architecture to Integrate Scarce-resources Wireless Sensor Neworks into Ubiquitous Environments

Pablo Pancardo[1], and Juan C. Dueñas[2]

[1] Universidad Juárez Autónoma de Tabasco, DAIS,
Cunduacán, Tabasco, México. C.P. 86690
pablo.pancardo@dais.ujat.mx
[2] Universidad Politécnica de Madrid, DIT-ETSIT,
28040 Madrid, España.
jcduenas@dit.upm.es

Abstract. Scarce-resources Wireless Sensor Networks (SWSN) are composed by a large number of sensor nodes with minimum processing capacity, short distance wireless communication and the ability to sense the environment. SWSN are ideal for gathering information from inaccesible or dangerous places in an autonomous way. Scarce resources impose conditions to the architectures, protocols and software in SWSNs, so, in general, ad hoc solutions are required and common network technologies can not fit into SWSN. However, it is neccesary for SWSNs to be integrated to other general-purpose networks to gain the benefits of information gathering by sensor nodes. Our proposal focuses on a network architecture design that makes it possible to integrate SWSN to TCP/IP based networks. In particular, we encourage interoperability as a fundamental attribute of the network architecture, so we propose the usage of mature, standards-based services gateways to meet this requirement.

1 Introduction

Advancement in microelectronic and the common tendency to create devices as small as possible has contributed to ubiquitous computing, so nowadays, tiny computers can be wirelessly networked and attached to almost any object. Embedded processors enabled with communication capacity and abilities to sense their environments feed their objects with the acquired information. Thus, Scarce-resource Wireless Sensor Networks (SWSN) are composed of nodes with capabilities to compute, communicate and sense the environment. Their nodes have limited resources in network bandwith, energy, simple short-range wireless communication, and very limited memory and storage. One of the principal limitations of a SWSN is the power supply in the form of a battery. Sensor nodes are intended to be physically small and inexpensive. Sensor nodes that compose a SWSN are able to provide with information that would be very difficult to obtain without their colective job. Environmental and critical infrastructures monitoring are examples. Technologies of computing, communication, information and integration, together with mobility properties and context-based services contribute to favour ubiquitous environments. SWSN help this goal offering information any time, anywhere.

However, SWSN's constrains impose conditions about protocols, software, and architectures that can be implemented into SWSN, becoming infrastructures where ad hoc solutions are required. Because TCP/IP technology requires high resources that are unavailable in SWSN, it ca not fit into them. Furthermore, TCP/IP philosophy is based in identity and node-centered communication while SWSN are data-centered. Aditionally, in SWSNs all operations have to be realized considering the maximum energy saving. All these situations have conduced to specific proposals for communications, processing, and network organization. In this way, we are considering a system architecture where SWSN can be integrated to TCP/IP based networks.

2 Scarce-resources Wireless Sensor Networks

Wirelesss Sensor Networks are composed of nodes distributed in a sensing area where each one has the capabilities to collect data and route them through radio channels until they reach a sink node wich could communicate with external networks [1]. A sensor network establishes a structure where it is possible to collect, process, analize, and diseminate data.

Particularly, a SWSN is concibed to be formed of constrained resources nodes wich must be small, therefore, they are limited in power, memory and processing capacity. For that reason, saving energy to prolong network life has a deep impact into the network architecture.

SWSN's magnitude could be in order to thousands of nodes and they have to meet some requirements as: self-configuring to establish connected networks despite failures of nodes (wich is common in this kind of networks) and dynamic environments, routing schemes where data attributes direct the path to follow instead of nodes addresses as in TCP/IP, in-networking processing to avoid waste of energy, and mechanisms to produce information as a result of a colective job of nodes.
SWSN's potential applications are envisioned to be impressive and revolutionary [2]. Some examples include: business and industrial automation where SWSNs can be implemented in conjunction with RFID (Radio Frecuency Identification) technologies having deep impact in areas like supply chain management; in-home elder healthcare system that monitors patients' medication intake; environmental monitoring specially suited for dangerous and risky places; and home automation.

3 Work in Progress

We have established our project's objective as a network architecture design to ensure interoperability between SWSN services and services implemented in TCP/IP-based networks. It includes to propose and evaluate SWSN's design patterns to carry out relevant quality requirements.

Our proposal considers the design of a service gateway with the required elements to integrate SWSN's applications with TCP/IP-based applications. The gateway must contain the effective software to obtain interoperability between the two kind of services. We have selected a Service-Oriented Architecture (SOA) because of SWSN's event-oriented nature and SOA's capabilities to provide good interoperability technology-wise overall, allowing services and applications built in different languages and deployed on different platforms to interact [3]. To accomplish our purposes we have defined some management services that include: basic information from the sensor network (device description, topology of the sensor nodes); synchronization; power management; and recording of utilization rate of the nodes. The System architecture is showed in Fig. 1.

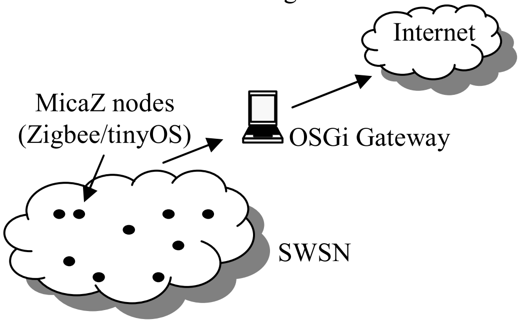

Fig. 1 System Architecture

The lowest level is constituted by the SWSN which is composed by Crossbow's Micaz nodes [4] (2.4 Ghz frecuency, maximum date rate 250 kbps, 128 KB program flash memory, 512KB measurement log memory, ~ 1 year lifetime, 75-100m outdoors range, 20-30m indoors range, 30 µW power comsumption during sleep, 33 mW while active), which are Zigbee [5] compliant; these kind of nodes are the most accepted worldwide for SWSN. These nodes work with the TinyOS Operative System [6] developed by University California Berkeley using the NesC [7] programming language, which is a reduced variant of C language. NesC was specifically created to be used in constrained-resources devices. A service gateway implemented over an OSGi Service Platform [8] is located in the middle of the system architecture and it has an integration function between SWSN and TCP/IP-based platforms. This gateway is used to interoperate with TCP/IP applications too. We consider OSGi R4 an excellent option due to specifications defined by the mobile expert group to tailor and extend OSGi for mobile devices that are data-capable, and also capable of connecting to wireless networks [8]. Overlay solutions have been ruled out due scarce resources.

We have selected free open source tools that let us to build scenarios where to validate our proposal. These tools will validate the overall system quality attributes, specifically interoperability. SWSN applications are implemented over the TinyOS operative system and can be simulated using TOSSIM [9], a TinyOS Network Simulator developed by UC Berkeley. Tinyviz is a graphical tool included in TinyOS that can be used to visualize communication parameters in TinyOS-based networks. A free TinyOS Plugin for Eclipse [10] has been used as Integrated Development Environment.

We consider developing some simulation scenarios for validating our proposal versus quality characteristics defined in a Product Quality Model ISO/IEC 9126-1 [11], where some metrics are provided to evaluate a software products. In our case we have taken the standard as a reference for the evaluation of system-wide characteristics rather than software quality. As cited in [3], the interoperativity attribute is an aspect of quality that should be supported in SOAs.

4 Conclusion and further work

The interoperability as quality attribute needs to be accomplished in any SOA. In this case, where we have heterogeneous platforms it has an essential importance. SWSNs seem to be revolutionary technology however, they can not be used in isolation and they need to be integrated to other technological infrastructures .

In our future work we will build scenarios and later we will evaluate them. Moreover, we will need to corroborate if the quality characteristics have been met.

References

1. Tubaishat, M., Madria, S.: “Sensor Networks: An Overview”, *IEEE Potentials*. April-May 2003. pp. 20-23. Volume: 22, Issue: 2. ISSN: 0278-6648.
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: “A Survey on Sensor Networks”, *IEEE Communications Maganize*, Vol. 40, No. 8, pp. 102-116, August 2002.
3. O’Brien, L., Bass, L., Merson, P.: Quality Attributes and Service-Oriented Architectures, Technical Note CMU/SEI-2005-TN-014, Carnegie Mellon University, The Software Engineering Institute, (2005).
4. Crossbow Technology Inc. http://www.xbow.com/
5. ZigBee Alliance. Available in http://www.zigbee.org
6. TinyOS http://www.tinyos.net/
7. Brewer, E., Culler, D., Gay, D., von Behren, R., and Welsh, M.: nesC: A programming Language for Deeply Networked Systems. UC Berkeley WEBS Project. Available in: http://nescc.sourceforge.net/(2004).
8. OSGi Alliance. http://www.osgi.org/
9. Levis, P.: Simulating TinyOS Networks. Available in: http://www.cs.berkeley.edu/~pal/research/tossim.html (2003).
10. Schuler, R., Burri, N.: TinyOS Plug-in for Eclipse. Distributed Computing Group, Swiss Federal Institute of Technology. Available in: http://www.dcg.ethz.ch/~rschuler/ (2006)
11. International Standard ISO/IEC 9126-1, Software Engineering-Product Quality Part 1: Quality Model. First Edition 2001-06-15.