Presence-Based Architecture for Wireless Sensor Networks Using Publish/Subscribe Paradigm

Ernesto García Davis, Anna Calveras Augé

Presence-based Architecture for Wireless Sensor Networks using Publish/Subscribe paradigm

Ernesto García Davis[1], Anna Calveras Augé[1,2]

[1]Department of Telematic Engineering, Universitat Politècnica de Catalunya
C/Jordi Girona 1-3, Mòdul C3, 08034 Barcelona, Spain
{ernesto.garcia,anna.calveras}@entel.upc.edu
[2]i2CAT Foundation, Barcelona, Spain

Abstract. Ubiquitous communication is a key component of Ambient Intelligence, enabling objects to communicate with each-other by means of a wireless ad-hoc network. A presence service allows knowing the availability or responsiveness status of elements in a communication (entities). Traditionally, only human use this service, however all smart devices or objects (with embedded wireless sensor nodes) could interact with each other thus a presence service could also enhance communication among them. To achieve this objective we look into existing protocols to propose the requirements needed to provide presence service on Wireless Sensor Networks (WSN). The main contribution of this paper is the discussion of requirements for presence services on WSN and the new proposed architecture to cope with presence services in WSN.

Keywords: Distributed Architecture, Wireless Sensor Networks, Presence Service, Embedded Devices

1   Introduction

Wireless Sensor Networks (WSN) consist of small distributed sensor nodes capable of communicate in a wireless network. WSN are providing tremendous benefits for a number of industries. In recent years, uses of WSN are moving from the mere monitoring of certain environmental variables towards its adoption for context aware systems. In this sense, WSN are able to capture a rich set of contextual information (e.g. spatial, physiological and environmental data) and to respond regarding to a situation without user interaction. In addition, context information could also contain another type of information called presence information. This information is related to user and communication conditions, capacity and preferences. Traditionally, presence information has been used by personal applications such as Instant Messaging (IM) in which human user can know the willingness or responsiveness of other human users to engage a communication. In a similar manner, presence information can also be used by other type of users such smart objects or devices in order to discover the status, conditions and capacity with the purpose of interacting with each other. In this paper, a "smart object or device"

refers to anything that has specific functionality and that is able to compute and communicate on its own. For instance, an energy meter can be programmed in a specific way. When consumption reaches a threshold it should send a command to turn off all devices with less priority, so we can save energy. Without presence information (such as on/off/standby/consumption level, user preferences, etc) the energy meter would try to send this command to all devices complying this condition. However, the energy meter would not know which of them are turned off. Therefore, unnecessary packets will be transmitted into the network resulting in waste of resources (energy, processing and bandwidth) at each node along the path to reach the destination device.

Some use cases to apply presence services on WSN are related to smart grid or Internet of Things. Firstly, Home Area Network (HAN) devices are capable of receiving messages from the Utility (service provider) and adjusting its operational mode based on Consumer preferences (e.g., energy saving mode, delayed turn on/off, etc). In these scenarios, a presence service could enhance decisions interacting between HAN devices and Utility equipment. Each HAN device will publish its presence information to the Utility equipment. The Utility will check the state of HAN devices presence before transmitting a command. In this manner, only HAN devices with availability status will receive a command, thus the Utility only will transmit necessary messages to the network. Secondly, the Internet of Things will be an open network in which auto-organized or intelligent entities will be interoperable and able to act independently pursuing their own objectives or shared ones, depending on the context, circumstances or environments. In the cases aforementioned, each object or device could incorporate an embedded sensor able to capture, to process and to communicate presence information through wireless media to enhance system performance.

Our motivation for this research is based on the fact that an efficient communication among users is possible if we know its presence information. Nevertheless, the presence service as we know it in Internet cannot be directly applied to the devices based on WSN. That is because these types of networks have constrained resources on processing, energy and bandwidth. The requirements in its architecture must be defined in an efficient manner.

Presence services, in general, are based on Publish/Subscribe communication model [1] [2]. Also, this model has been integrated in several protocols specifically designed for WSN (e.g., service discovery protocols and middleware technologies). However, as far as we know there is not a protocol or mechanism which considers the whole requirements to provide presence services in WSNs.

In this paper, we present the requirements to provide a presence service on WSN and then we propose a presence-based architecture using the Publish/Subscribe communication model. The remainder of this paper is organized as follows. In section 2, we provide background information about Presence Service. Followed, section 3 presents related work for Publish/Subscribe Protocols on WSN. In Section 4 we discuss the requirements to provide presence service on WSN and a new presence-based architecture for WSN. Finally, conclusions and future work are discussed in section 5.

2 Presence Service

Presence information is a subset of context information. It expresses the determining factors around communication between users or devices (such as their availability, willingness, environment, preferences, etc). The presence service architecture [3] fits into publish/subscribe communication model. This architecture works as follows (Fig. 1): the presence entities, called presentities (devices or objects), provide their presence information to the presence service. The presence service accepts, stores, and distributes the presence information to everyone subscribed to get the notification about presence changes. The entities (devices or objects) receiving notification about presence information are called watchers.
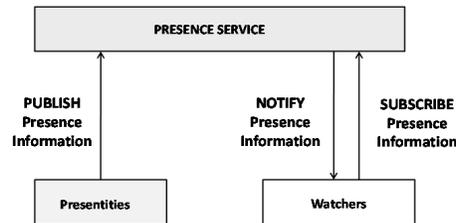

Fig. 1. General Presence Service Architecture

The presence information will improve the whole communication because watchers only will transmit messages to presentities taking into account their presence information. Therefore, we achieve an efficient communication on WSNs taking into account the constrained resources on this type of networks. Presentities and watchers communicate by exchanging presence events. Presentities are publishers in Publish/Subscribe Communication Model and watchers are subscribers that are interested in a presence event.

In the next section, we briefly present publish/subscribe protocols already known in the area of sensor networks. We focus on the issues they are addressing in their design.

3 Related Work of Publish/Subscribe on WSN

Several authors [11] [12] have proposed protocols making use of publish/subscribe communication model. In this section, we summarize these protocols focusing on relevant aspects for presence services on WSN.

3.1 Message Queuing Telemetry Transport for Sensor Networks

Message Queuing Telemetry Transport for Sensor Networks (MQTT-S) is an extension of Message Queuing Telemetry Transport (MQTT) [13] originally developed for telemetry applications using constrained devices. It is based on a

centralized architecture where a broker node is placed in the backbone network to provide publish/subscribe service to the nodes in this part of the network and in the WSN. MQTT-S allows subscribing to one topic and receiving information of several nodes by using the "wildcard" concept. Therefore we can reduce the number of transmitted messages to only one SUBSCRIBE message instead of sending one for each device. That results in energy saving for the subscriber nodes which is very relevant in large WSN. MQTT-S defines three QoS levels. Level 0 offers a best effort delivery service and no retransmission or acknowledgment is defined. Level 1 allows the retransmission of messages until they are acknowledged by the receivers. However, certain messages may arrive multiple times at the destination because of the retransmissions (duplicated). Finally, level 2 ensures not only the reception of the messages, but also that they are delivered only once to the destination. In addition, this protocol considers the message size to be no longer than 64 bytes of the payload of an 802.15.4 packet (127 bytes). That is because the remaining 64 bytes should be used by the overhead information required to support functions such as MAC layer, networking, security, etc.

3.2 Tiny Session Initiation Protocol

Tiny Session Initiation Protocol (TinySIP) is based on Session Initiation Protocol (SIP) protocol. It incorporates a messaging mechanism for accessing sensor-based services. It provides a communication abstraction which allows session signaling, publish/subscribe services and instant messaging features. Basically, this protocol is based on centralized architecture. A SIP-TinySIP Gateway(s) situated on the edge of WSN. All communication between clients on outsider networks and WSN nodes is through the SIP-TinySIP Gateway(s). In addition, TinySIP uses a more compact message size than MQTT-S. A TinySIP message can be transmitted within a small payload, such as the 29-byte of a regular TinyOS payload message.

4 Proposed Architecture

In this section, we firstly present the requirements for presence service. Next, we propose a new architecture based on publish/subscribe to cope with presence service on WSN.

4.1 Requirements for Presence Service on WSN

In general, presence services on Internet have certain limitations to be used on WSN. The most relevant are:

- Limited vocabulary to express presence information: In the case of human users, the presence information was contained in PIDF [4] format. It includes a simple status of presence information (e.g.: offline, busy, away), and this is not enough to convey a user's real situation and to decide how to establish a communication with an entity. Some authors have identified this insufficiency of basic

information and responded with several extensions of presence information [5] [6] [7] that are RPID, CPID and GEOPRIV respectively. However, to enable the integration with objects or devices, we need to define others extensions. Therefore, we can accommodate additional type of information provided by sensor nodes embedded on devices or objects. A first approach to extend the vocabulary is presented in [8] where authors propose the addition of two new attributes to the GEOPRIV. Nowadays, there is not yet consensus in the used data model to describe the presence information of objects or devices.

— The presence status is generally updated manually by the user: There is not a way to deduce automatically presence status of a user. Some presence technologies can change presence information status to "away" assuming that there is not activity from the user in the device during some period of time. In the case of objects or devices acting as presentities, it should be necessary that they change their status without user interaction. That can be made by means of context information acquired by sensor nodes.

Moreover, there are other requirements we have to consider based on the specific characteristics of WSN such as:

— Reduced size of MTU: the size of presence information messages used by traditional presence services is incompatible to fit into the data packets used in embedded devices with constrained resources, such as sensor nodes. It is needed an encoding/compressing mechanism to convey presence information on WSN.

— Limited resources in terms of energy, computing and storage: The availability of these resources depends on the number of transmissions and the length of transmitted packets. Authors in [11] [12] do not provide a mechanism to manage resources in an efficient manner. In this case, we could apply data aggregation techniques [9] that allow, for example, concatenating individual data items into larger packets thus reducing network traffic load and saving energy.

— Wireless communication: WSN are susceptible to loss packets due to transmission errors. If presence information cannot be delivered to watcher nodes, the communication among entities could be affected. Therefore, it is necessary to implement a reliable data delivery. In this case we could consider using Transmission Control Protocol (TCP). However, the header overhead, connection management, end-to-end flow and congestion control lead to TCP poor performance on wireless environment and resource constrained nodes, such as WSN [10]. This issue is still opened and some authors [11] suggest assign this functionality to the application layer.

4.2   General Description of the Proposed Architecture

Taking into account the relevant features of current publish/subscribe protocols and the requirements above mentioned, we propose a new approach for a presence-based architecture. There are three entities that interact in our architecture. The first is a publisher entity that is used to produce presence information and publish it when it is requested. Second, there is a subscriber which is the entity that makes a subscription

in order to receive presence information for its interest. Finally, there is a broker entity responsible for distributing the published information from publishers to subscribers. Our architecture, contrary to centralized architecture such [11], is based on distributed architecture.

In centralized architecture there is only one broker node located between WSN and the external network. So, the broker node makes the sink node functionality. Therefore, when many nodes are forwarding messages to the sink node at high rates, it could become a bottleneck leading to low network performance. In addition, this architecture is unsuitable for providing a presence service on WSN because these protocols consider the communication will only occur between sensor nodes and external users or vice versa. Thus, all the information is sent to the broker node (sink) located between WSN and external network. However, a presence service on WSN and new trends such as "Internet of Things" consider that objects can interact among them. Thus, publish/subscribe protocols should consider publisher nodes not always will send information to the sink node as the only destination. We propose a distributed architecture, where there are other broker nodes inside WSN aside from the broker node acting as sink node. We can see this distributed architecture in Fig. 2 where there are, as example, two broker nodes inside WSN, and another broker node at the edge of the network. This one is also acting as gateway to communicate with the external network. Thus, it allows distributing the load in the network by grouping publisher and subscriber nodes in different broker nodes.
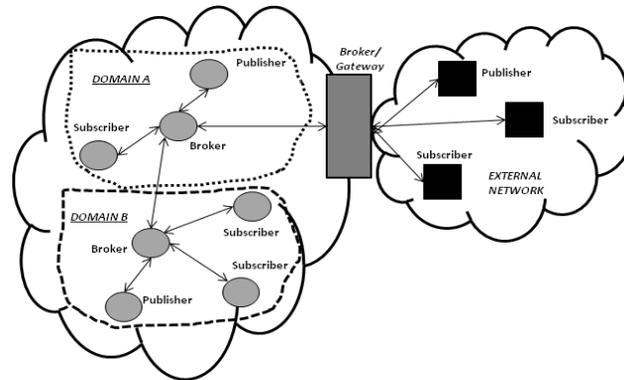
Fig. 2. Proposed Presence Service Architecture for WSN

We call *broker domain* to the group of subscribers and publishers nodes managed by a broker node. The broker nodes will be the only responsible to exchange message between broker domains, as we can see in Fig. 3. When a potential publisher or subscriber node enters into the network, it sends a broadcast message to discovery a broker node in its domain. Once the publisher or subscriber node receives a reply, it will establish a connection with the broker node.
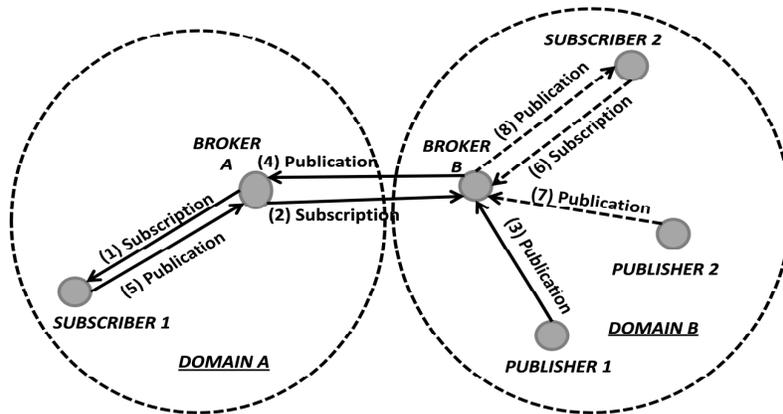
Fig. 3. Broker roles and broker domain

Also, a broker node might play a publisher or subscriber role on behalf of its domain. Fig. 3 shows this situation. "Subscriber 1" and "Publisher 1" are placed in different broker domains. "Subscriber 1" on broker domain A wants to receive presence information from "Publisher 1" on broker domain B. So, "Subscriber 1" makes a subscription (1) in its respective broker node ("Broker A"). This one makes a subscription (2) in the broker node responsible for broker domain B ("Broker B"). When "Publisher 1" sends presence information (3) to its broker node, this one will forward it (4) to broker node of Broker Domain A, and this one; will transmit it (5) to the "Subscriber 1". In this way, nodes located in different Broker's domains will exchange messages through corresponding broker nodes. It should be noticed that a broker node learns the identities of brokers in other domains. That is because it receives a broadcast message of available topics to subscribe in each broker. Then, it will store this information to be available when a potential subscriber node wants to subscribe to a topic. On the other hand, publisher and subscriber nodes located in the same broker domain, will exchange messages through the broker node responsible for this domain. Thus, exchanged traffic is kept inside the broker domain. This situation is also depicted on Fig. 3, where "Subscriber 2" makes a subscription (6) through "Broker B" to receive presence information from "publisher 2" on the same broker domain B. When "Publisher 2" publishes presence information (7) to its broker node, "Broker B" will forward it (8) to "Subscriber 2". In conclusion, the proposed architecture contributes to isolate the exchanged traffic in broker domains, and will exchange traffic between broker domains through corresponding broker nodes only when it is needed.

The proposed architecture not only improves scalability but also provides mechanisms for an efficient energy management. Moreover, it reduces the subscriber nodes spent time to receive the first publication. These mechanisms are explained in following sections.

4.2.1 Publication on Demand of Presence Information

Generally, publish/subscribe protocols allow a publisher to publish data regardless of the presence of a subscriber. This would result in extra power consumption at publisher node. Moreover, unnecessary network traffic increases the risk of network congestion. To overcome this situation, in our architecture, publisher nodes only send publications when there is a subscriber. So, the broker sends a notification message to the publisher the first time it receives a subscription. That is shown in Fig. 4.
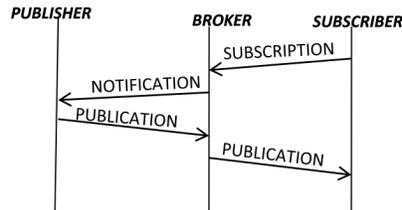

Fig. 4. Publication on demand

Also, similar notification message is sent to publisher when the last subscriber decides unsubscribe to presence information event (un-subscription). In this case, the publisher node stops events of presence information publication. This mechanism is referred as publication on demand of presence information. It should be noted that previously publishers communicate to the broker the events they are able to publish. This way, a broker knows about the publishers to send them the notification of subscription.

4.2.2 Data Aggregation

As we mentioned previously, data aggregation is a well known and essential technique to achieve energy efficiency when propagating data from WSN nodes. In our architecture, we make use of wildcards in similar way as MQTT-S [11] does it. This allows that we can transmit only one SUBSCRIBE message to subscribe us to a topic related to several publishers. In addition, we propose two ways to perform data aggregation with the purpose to decrease the number of messages into the network. Data aggregation will be performed either in the publisher or in the broker node.

- *Data aggregation in publisher node*: we propose the publisher nodes can aggregate packets in a "lossless way". That means that publisher will not transmit publications to the broker node until a timer expires. They will aggregate in a single message all publication events produced during the timer period. Then, they will create only one packet to transmit to the broker node. This technique results in reducing the number of transmissions and thus energy saving of network nodes. However, there is a trade-off between the number of transmissions and the delivery delay. The more the number of gathered data; the less is the number of transmissions. Nevertheless, waiting for more data increases the delivery delay. This issue is addressed by the new proposed

architecture. Therefore, we recommend this technique only be applied in situations where network congestion is detected or an application parameter has been defined. We can see the algorithm in Fig. 5 where a publisher node detects a congestion indication (i.e. received from the broker or from MAC layer) or there is an application parameter defined. Then, the publisher node will activate a timer, we call "aggregation timer". During this period, publication events produced locally will be buffered and marked for "lossless" aggregation.
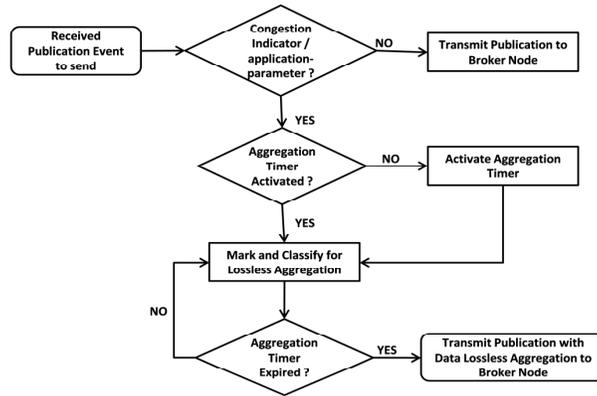

Fig. 5. Lossless Aggregation Algorithm

The number of packets to be buffered will depend on memory resources. Also, the number of data that one packet can carry will depend on the packet size limit. For example, 127 bytes is the maximum size of PHY service data unit defined in IEEE 802.15.4 standard. When the "aggregation timer" expires, the publisher node performs the lossless aggregation, and transmits the packet to the broker node. In case of congestion network, this mechanism prevents the publisher nodes continue transmitting packets. Thus, it avoids the increment of network congestion. Furthermore, it contributes to save energy. That is because of, during congestion, the transmission of packet would result in packet loss, thus extra packet retransmission.

— *Data aggregation in broker node*: The broker node aggregates publication of concerning publishers (i.e. by configurable function: average, min, max, status, etc). Then, it transmits only one packet to the subscriber nodes.

4.2.3 Advertisement of presence information to subscribers

Another important issue in presence services over WSN is that subscriber nodes receive presence information as soon as this has been published. The proposed architecture allows publisher nodes to include presence information (initial value) when they register the topics at the broker node as is showed in Fig. 6(a). This presence information will be stored in the broker node and will be available immediately in the subscriber.
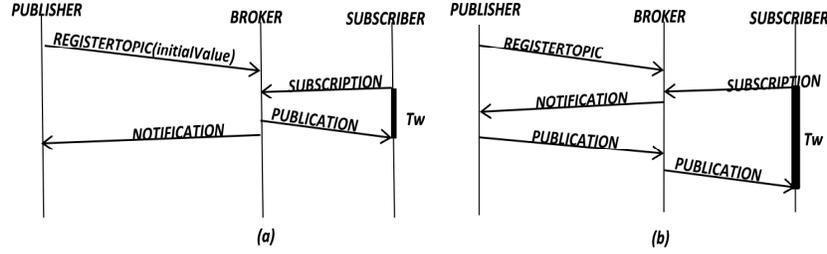
Fig. 6. Spent Time in the first publication

Let suppose node 1 is the first subscriber for the registered topic A. Now, let consider T_w as the waiting time experimented by subscriber node 1 to receive the first publication. T_{pb} is the time used by the broker node to process the subscription of topic A and transmit the publication message to subscriber node 1. Then,

$$T_w = T_{pb} \tag{1}$$

Otherwise, if publisher node does not register topic with initial value as it is showed in Fig. 6(b), T_w would be calculated by,

$$T_w = T_{np} + T_{ep} + T_{pp} + T_{pb} \tag{2}$$

Where T_{np} is the time used by the broker node to send a notification message indication to publisher to initiate the publication for topic A; T_{ep} is the time until the next event for topic A is produced; and T_{pp} is the time used by publisher node to generate publication message and transmit it to the broker node.

4.3   Evaluation

A first glance to the mechanisms provided by the proposed architecture can indicate several valuable advantages. It reduces the traffic in the whole network by isolating it inside broker domains. That is an important issue in large WSN. Moreover, it addresses the energy efficiency applying data aggregation techniques. Also, the architecture contributes to reduce the spent time of subscriber nodes to receive publications. Simulations were carried out to make a preliminary evaluation of the proposed architecture. We use OMNet++ [14] simulation environment to perform several experiments. The network topology consisted of one broker node, one subscriber node and up to 32 publisher nodes depending on test scenarios. Each node is provided by an 802.15.4 network interface working in 2.4 GHz frequency. We have analysed two main aspects. Firstly, we measured the spent time of subscriber nodes to receive publications. In this case, Fig. 7 shows that the mechanism described in section 4.2.3, where publisher node is able to include initial value when it registers a topic reduces the waiting time around 38% comparing if we don't use this. Secondly, we evaluated the relation between the impact of the amount of publication messages without subscription in the network traffic when we increase the number of publisher nodes and the energy consumption when we increase the publication rate.
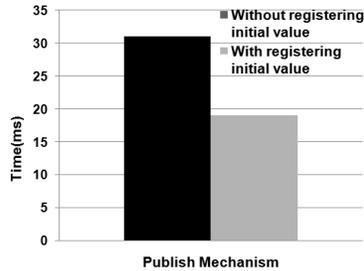
Fig. 7. Spent Time until reception of the first publication by subscriber node

From Fig. 8 we can note a proportional direct relation between the number of publish messages transmitted without subscription and the number of publisher nodes.
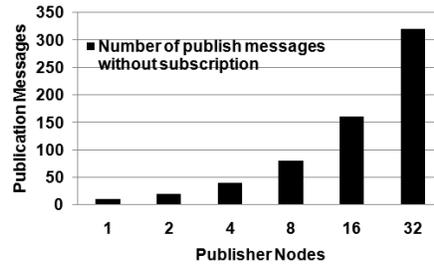
Fig. 8. Publish messages transmitted without subscription

In the simulations, each publisher node has a publish rate equal to 10 messages per second with a regular inter-packet interval. As we mentioned previously, this way to work has a high impact on energy resource of nodes and also generates unnecessary traffic on the network. Therefore, the advertisement of presence information to subscribers has a great benefit. Fig. 9 shows that our approach described in section 4.2.1 results in an energy saving between 3.6% for the lower publication rate and 80% for the higher one. There is an important benefit of using publication on demand mechanism.
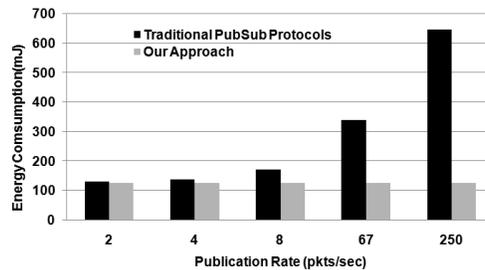
Fig. 9. Energy Consumption of Publications on Demand

5   Conclusions

In this paper, we analyse the implementation of presence service in WSN. We discuss its requirements and present a new architecture to provide presence services in WSN. The mechanisms provided by our architecture bring significant advantages with respect to other publish/subscribe protocols in WSN. The publication on demand mechanism prevents to transmit publication messages without related subscriber nodes. In addition, this architecture provides a mechanism to avoid that subscriber node waits for the next presence information produced by publisher node. It reduces waiting time of subscriber nodes. Preliminary results presented in this paper show that the proposed architecture is suitable for WSN. The implemented "lossless" data aggregation algorithm would result in energy saving for publisher nodes. Moreover, it would reduce the traffic load in situations of network congestion. At the present time, more results are being obtained by simulation to quantify how much energy can be saved if we apply "lossless" data aggregation. Also we intend to concentrate our efforts in building and testing a hardware/software prototype.

References

1. Eugster, PT. The many faces of publish/subscribe. ACM Computing Surveys. June 2003.
2. Tran, Duc A. et al. Publish/Subscribe Techniques for Sensor Networks. May 2009.
3. Day, M. et al. Presence and Instant Messaging Model (RFC2778). February 2000.
4. Sugano, H. et al. Presence Information Data Format. PIDF (RFC 3863).Aug. 2004.
5. Schulzrinne, H. et al. Rich Presence Extensions to the PIDF (RFC 4480) July 2006.
6. Schulzrinne, H. et al. Contact Information for the PIDF (RFC 4482) July 2006.
7. J. Peterson. GEOPRIV Location Object Format. (RFC 4119) December 2005
8. El Barachi, M.et al. A Presence-Based Architecture for the Integration of the Sensing Capabilities of WSN in IM Subsystem. IEEE WCNC 2008. 1525-3511.
9. Abdelzaher, T. et al. Feedback Control of Data Aggregation in Sensor. 43rd IEEE Conference on Decision and Control, 2004. 0191-2216
10. Kuorilehto, M., et al. Experimenting TCP/IP For Low-Power Wireless Sensor Networks. IEEE PIMRC'06.Sept. 2006. pp 1-6.
11. Hunkeler, U. et al. MQTT-S. A publish/subscribe protocol for Wireless Sensor Networks. COMSWARE 2008. Pages. 791-798.
12. Krishnamurthy, S. TinySIP: Providing Seamless Access to Sensor-based Services. 3rd International Conference on Mobile and Ubiquitous Systems - July 2006.
13. MQTT, http://mqtt.org
14. OMNet++ Simulator. http://www.omnetpp.org