Locality and Interest Awareness for Wireless Sensor Networks (LIASensor)*

Bruno Andrade, Luís Veiga and Paulo Ferreira

INESC-ID, Instituto Superior Técnico, Lisbon, Portugal
`bruno.jesus.andrade@ist.utl.pt`, `luis.veiga@inesc-id.pt`, `paulo.ferreira@inesc-id.pt`

Abstract. Wireless Sensor Networks (WSN) consist in spatially distributed set of autonomous sensors along an area to be monitored. Such WSNs can be used both in military and civilian industry. Whereas in the military field, WSNs are used to monitor areas, that otherwise could not be possible to monitor, in civilian industry their usage covers various fields, such as health, farming, and environmental monitoring. WSNs can monitor vast extensions; however, there are limitations regarding their usage, being that energy consumption is the biggest problem.
Power consumption is directly related with communication distance and message's length. This work presents **L**_ocation and_ **I**_nterest_ **A**_wareness for WSNs_ (**LIASensor**), a WSN architecture for environmental monitoring. LIASensor reduces the network traffic through interest and locality awareness techniques thereby increasing the whole network lifetime. LIASensor was implemented and simulated in Castalia Simulator with promising results.

Keywords. Wireless Sensor Network, Interest Management, Locality-Awareness, Interest-Awareness

1 Introduction

Nowadays, sensor nodes are used worldwide in a large range of applications. Due to technology advances [3, 4] sensor nodes have become powerful, cheap, small and consequently disposable, which allowed the emergence of Wireless Sensor Networks (WSN). WSNs are composed by a large number of scattered sensor nodes that must be able to self-organize and forward information collected back to the end-user.

The biggest problem regarding WSNs is the scarce power source, typically a limited battery, and it is well known that sensors consume more energy when they are transmitting information than when they are collecting and processing data [11]. Furthermore, power consumption increases as the message's length, and the communication distance increases. With regard to power consumption any extra bit or extra meter counts since the death of few nodes may create blind spots, or in the worst case scenario, the death of the entire network. So, at this moment, developments and new techniques capable of reducing both the number and size of messages are necessary.

The main goal of this work is to reduce the number of messages within a wireless sensor network, and thus increase the lifetime of the entire network. This work focuses in environmental monitoring by foot, in scenarios on which the evolution of the situation is important. For instance, in volcanic or seismic region where sensors must report more information than in other kinds of environmental monitoring, since these regions can change dramatically fast. In this type of monitoring, a user patrols an area with a portable device that transmits queries requesting information from sensors.

This work proposes an improvement for these scenarios: LIASensor is able to reduce both messages' size and number; through **Interest-awareness** LIASensor reduces the number of messages, and through **Locality-awareness** the messages' size.

This document is organized as follows. In section 2 we describe the state of art regarding Wireless Sensor Networks, and methods that are able to reduce messages in Ad-hoc networks. Then, in section 3, and 4 we present respectively the architecture and implementation of LIASensor, at last, in section 5, we evaluate and analyze the results of the system prototype

2 Related Work

WSNs may vary a lot; there are many factors influencing and changing these networks. Over the years, many protocols and algorithms have been proposed, making these networks more complex or simpler.

This section describes the state of the art of Wireless Sensor Networks. Firstly, it is presented some fundamental concepts, followed by WSN classification and wireless sensor protocol stack. Secondly, it is presented a brief section about energy consumption. Then, it is introduced some cluster-based routing algorithms. And at last, it is presented methods that reduce the number of messages in ad-hoc networks.

2.1 Fundamental Concepts

A sensor is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. The output can be immediately read or be transmitted electronically over a network for reading or further processing. Sensors are used in everyday objects such as touchscreens displays (tactile sensor) and lamps which dim or brighten by touching the base.

According to [13], a WSN typically consists of a large number of low-cost, low-power, and multifunctional sensor nodes that are deployed in a region of interest. These sensor nodes are small in size, but are equipped with sensors, embedded microprocessors, and radio transceivers, and therefore have not only sensing capability, but also data processing and communicating capabilities. They communicate over a short distance via a wireless medium and collaborate to accomplish a common task, for example, environment monitoring, battlefield surveillance, and industrial process control.

2.2 Classification of WSN and Protocol Stack

Wireless Sensor Networks are applications specific; usually, they are deployed for particular applications, so different networks have different characteristics. According to different criteria, WSNs can be classified into different categories [13]. For instance, WSNs may be static, mobile, deterministic, nondeterministic, single-hop, multihop, etc.

Like regular networks, WSNs also have a five layer protocol stack [1]: *physical layer*, *data link layer*, *network layer*, *transport layer*, and *application layer*.

In addition to the five layers, the protocol stack can be also divided into three groups of management planes across each layer, including power, connection, and task management planes.

2.3 Energy Consumption Models

Accurate and low-cost sensor localization is a critical requirement for the development of wireless sensor networks in a wide variety of applications. There are some techniques suited to locate nodes; however, all of them have pros and cons. [10] described measurement-based statistical models useful to describe many of these methods.

Among the location techniques, these are the most used:

- **Time of Arrival (TOA)** - Calculates the distance through time and signal propagation velocity. It is the travel time of a radio signal from a single transmitter to a remote single receiver. Since TOA relies on the difference between the time of arrival and time of departure, all receivers and transmitters must be synchronized so there is no error in the difference due to clock offsets. This may prove to be a problem, especially considering the high speed at which the signals travel. Also, as with any time sensitive systems, there is also the possibility of significant hardware delays that must be accounted for to calculate the correct distances.
- **Angle of Arrival (AOA)** - Calculates the distance by getting the signal direction send by the adjacent node through the combination of array antenna and multiple receivers;

- **Received Signal Strength Indication (RSSI)** - Calculates the distance through the strength of the received signal.

A global positioning system (GPS) receiver on each device is a good solution to the future, but at this moment it still monetary and energy prohibitive for many applications.

2.4 Clustering algorithms

Routing protocols are one of the most important components of WSNs; these protocols are responsible for identify, select and decide which paths are the most energy-inexpensive, direct and reliable; without these protocols WSNs would not be feasible. According to Al-karaki [2], routing protocols in WSNs can be divided into three categories: **flat-based routing**, **hierarchical-based routing** and **location-based routing** depending to network structure.

Among these three types of protocols, hierarchical-based or clustered-based is the most energy-inexpensive. Moreover, the utilization of clustering algorithms provides 1) scalability enhancement, 2) communication efficiency and 3) possibility of data aggregation.

In cluster-based routing protocols, sensors are organized in regions known as clusters. Each cluster has a single cluster-head (CH), and many cluster-members. Cluster-members (just called sensor nodes) send their information to cluster-heads, which forward the information collected to the sink or base-station. Each node assigns itself to one cluster-head. Depending of which algorithm is being executed; the number of clusters might be very different, yet zero cluster-heads or 100% of cluster-heads is the same as direct communication. The number of cluster-heads can be assigned directly before the network being deployed or dynamically during the routing algorithm execution.

Cluster algorithms communications are made through the following two modes:

- **Intra-cluster communication** (forwarding to cluster-head): each sensor node sends the sensed data to the elected cluster-head;
- **Inter-cluster communication** (forwarding to base-station): each cluster-head sends data either to neighboring cluster-heads or directly to the sink whether it is near.

2.5 Message Reduce Techniques in Ad-Hoc networks

Besides enhancing the network lifetime through the utilization of efficient routing protocols, it is absolutely necessary to reduce message's length and number too. To achieve this, some methods and techniques have been used in WSNs. Data fusion [6] is currently used in WSNs; Vector Field Consistency [12] is a more recent model, but with great prospects.

Interest Management *Interest Management* (IM) is a set of techniques able to filter and dispose relevant information to entities interested in it. This method can be abstracted using *publish-subscribe* models [5]. In such models, *Publishers* are objects that produce events and *Subscribers* are objects that consume events, whereas an object can be both a publisher and a subscriber. Without *Interest Management*, a receiving entity would receive messages from every producing entity; so, the receiving entity would be responsible for sorting through and discarding useless messages. The concept of Interest Management was developed to address this problem by reducing the arriving messages to a smaller and relevant set. Under Interest Management, an entity expresses its data interests in terms of location and other application-specific attributes. According to Morgan [9], other agents in the simulation infrastructure, *interest managers* (IM), accept entities' *interest expressions* (IEs) and use them to filter messages to sets (or reduce supersets) which meets the entities' needs. An *Interest Expression* (IEs) is a specification of the data one entity needs to receive from other entities in order to interact with them correctly. IEs may refer to several attributes of the simulation entities or to the radius of interest, i.e. an entity may express interest about all entity within a 50 meters radius around itself.

Vector Field Consistency Vector Field Consistency (VFC) [12] is a client-server architecture based on interest and locality awareness techniques. Furthermore, VFC is an optimistic consistency model allowing bounded divergence of object replicas.

In VFC, the server is the responsible for keeping the actual state of the world, and for regularly update clients. Each client registers within the server the objects, also called pivots, that will be shared, as well as, their consistency parameters, which consist in a 3-dimmensional vector.

A pivot generates a consistency field determining the consistency of each object as a function of the distance between the object and the pivot. The pivot's surroundings are updated according with its consistency degree.

Consistency degrees are defined through 3-dimmensional vector, which specifies: 1) the maximum time a replica can be without being refreshed with its latest value, 2) the maximum number of lost updates replicas, i.e., updates that were not applied to a replica, and 3) the maximum relative difference between replica contents or against a constant.

Data fusion Data fusion, also called, data aggregation is the process of integration of multiple data and knowledge representing the same real-world object into a consistent, accurate, and useful representation. The expectation is that fused data is more informative and synthetic than original inputs.

Applying data fusion to ad-hoc networks, specifically WSNs is very beneficial to the whole network due the decreasing of redundant information. Sensor nodes are usually deployed in large or even huge number in systems or areas of interest. Because of dense pattern of sensor deployment, neighboring sensor nodes may sense similar data on specific phenomenon. Since sensor nodes are run by battery power, it is critical to perform every operation in an energy-efficient manner. For this purpose, it is desirable for a sensor node to remove the redundancy in the data received from its neighboring nodes before transmitting the final data to the sink. Data aggregation is an effective technique for removing data redundancy and improving energy efficiency in WSNs. The basic idea is to combine the data received from different sources so that the redundancy in the data is minimized and the energy consumption for transmitting the data is reduced. The data-centric nature of WSNs makes data aggregation a crucial task [7]. Different authors consider different fusion methods. Li [8] considers five representative fusion methods. However, Zheng [13] considers there are only three major data aggregation techniques.

Over the years many solutions have been proposed to increase WSNs lifetime; however none of them really tried to reduce the number and the size of messages within WSNs. The majority of solutions aimed data routing optimization, neglecting the number and the size of their messages. Other methods, such as VFC are well suited for messages reduction, but the adaptation to WSNs is not straightforward. Data fusion is already used in WSN; however it is useless in environments where sensor nodes are far away from each other.

3 Architecture

This work proposes a new approach in Wireless Sensor Networks; LIASensor reduces the number and size of messages in networks through Locality and Interest awareness techniques.

LIASensor architecture was been designed for constant changing environmental monitoring, and its main properties are locality and interest awareness. Considering sensor networks protocol stack (subsection 2.2), LIASensor is an application layer technique. As a result, LIASensor uses bottom layers API, which permits it to be executed on top of any routing or mac protocols. The rest of this section presents in more detail both LIASensor architecture and operation mode.

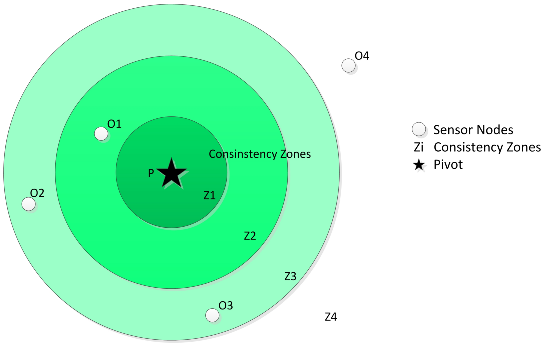

Fig. 1. Conceptual consistency zones.

3.1 Client-Server Architecture

The model of communication used by LIASensor is a client-server architecture. In This kind of architecture servers provide functions or services. On the other hand, clients are responsible for initiating communications with servers, requesting services or functions. In LIASensor every node plays a single role; the sensor nodes act like servers whereas the user-ends acts like clients.

Further, LIASensor only considers two relationships in this architecture: **many to one**, and **many to many**. This means that in any point of time, it is possible the existence of one or more end-users requesting data from the environment. On the other hand, sensor nodes must always exist in larger numbers.

3.2 Locality-awareness

To accomplish Locality-awareness the end-user generates radius of interest around him, which are formed through the device's signal strength. The user transmits his query directly to nodes, without any forwarding; nodes that do not receive any query are considered outside of the radius of interest, and so, these nodes do not respond. If the end-user wants to know any information outside his radius of interest, he must increase his device signal, or move through another direction.

LIASensor, by analogy with the electric $\vec{E}$ and the gravitational $\vec{G}$ fields generates consistency fields determining the consistency of each sensor node as a function of the distance between the sensor and the pivot. Thus, pivots generate consistency zones, iso-surfaces, ring shaped, concentric areas around them, such that the objects positioned within the same consistency zone are enforced the same consistency degree, For example in Figure 1 a user P is in the center of four consistency zones labeled z_i, where $0 \leq i \leq 4$. Objects o_2 and o_3 are enforced the same consistency degree since they are in z_3.

Sensors in pivot's surroundings respond according with their consistency degree, a node closer has a higher consistency degree than a node further, and therefore its response message is more detailed and complete. Sensors know on which consistency degrees they are by calculating the distance between them and the user.

3.3 Interest-awareness

WSNs contain many sensor nodes, and these sensors may have different sensing abilities. For instance, they might be skilled to sense temperature, humidity, pressure, seismic activity, presence of humans or animals, among others environmental attributes. Therefore, the end-user must be able to request only appropriate information. This is accomplished before query transmissions. The end-user's device is able to transmit specifics requests throughout the network. Upon receiving the query, it is up to each sensor to decide either if the received query matches its sensing functions or not. Only sensors capable to respond coherently reply their response back.

3.4 Finite State Machine

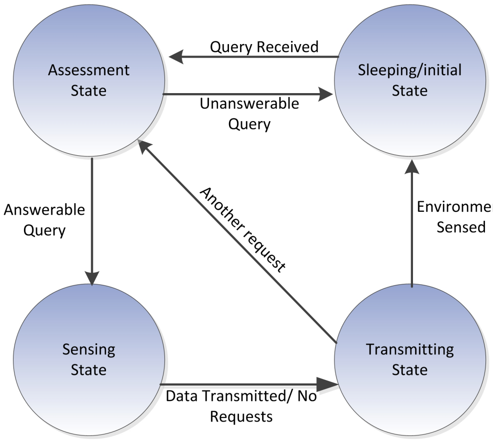

Fig. 2. State Machine Representing the sensor nodes.

LIASensor works as a finite state machine. Each sensor is at one state at any given time. The list of states contains four different states (sleeping, assessment, sensing and transmitting states), and the list of inputs contains six elements (query received, unanswerable query, answerable query, no requests, another request and environment sensed).

Considering the FSM (figure 2), every sensor begins at the sleeping state (the initial state). Upon the occurrence of the event (the query arrival) the sensor switches from the sleeping state to the assessment state. At the assessment state, the sensor checks whether the query is addressed to it or not. The assessment is based in both interest and locality awareness techniques. If the assessment returns a positive value, i.e. the sensor can answer the query, the sensor transits to the sensing state. Otherwise, the sensor gets back to the sleeping state.

At the sensing state the sensor senses and stores environmental data. When the sensing task is completed, the sensor switches from its sensing state to the transmitting state. At the transmitting state the sensor constructs a packet compound by the actual sensed data and its previous records, and then sends the information to the end-user. When it finishes the transmissions task, the sensor has two different paths regarding the triggering event, If a new query event is triggered the sensor get back to the sensing state, otherwise the sensor finishes the cycle and returns to the initial state again (the sleeping state). Table 1 summarizes the FSM operation.

Current State	Input	Next state
Sleeping State	Query received	Assessment State
Assessment State	Unanswerable query	Sleeping State
	Answerable query	Sensing State
Sensing State	No requests	Transmitting State
Transmitting State	Environment sensed	Sleeping State
	Another request	Assessment State

Table 1. LIASensor state transition table.

4 Implementation

LIASensor is implemented over Castalia[1], a realistic wireless and radio modeling simulator for Wireless Sensor Networks. Castalia is based in OMNeT++[2] platform and develop in C++. LIASensor algorithm is an application layer protocol also developed in C++.

The rest of this section describes the solution's enforcement.

4.1 Locality and Interest Awareness Enforcement

As subsection 3.2 presented, locality-awareness uses radius of interests as its main element. The analysis of radius of interest encloses sensors within a consistency degree, which states the responses size.

Every query has a RSSI associated to it, which is used to calculate the distance between the user and the sensor. When a query is transmitted, the nodes that sensed the query read the querys RSSI and compare it with their consistency view, a table that associates RSSI values into consistency views. Consistency views are detailed in subsection 4.2. If the RSSI matches one of the tables rows, the response is constructed and sent with the correct amount of information; otherwise the query is ignored and dropped.

The enforcement of interest-awareness is easily accomplished. The strict comparison between the request type and the sensor skill is sufficient. Every sensor node is skilled to perform some task and every user request has a field that identifies the information on which the user is interested. So, when a query arrives, the sensor node captures the packet, read the type field and compares it with its function. If the type field matches its function a new response is sent, otherwise the request is ignored and dropped.

4.2 Consistency View

Sensor nodes have a consistency view, which can be described as a table that contains the interval of RSSI values mapped into consistency degrees. Each consistency view row is composed by: a) a RSSI interval; b) a consistency degree; and c) the amount of information a sensor must forward when a query arrives. The amount of information forwarded should increase as the consistency degree increases too. In other words, higher consistency degrees mean more information.

Values within the consistency table might be replicated without affecting the way consistency views work. When a query arrives, sensors search for matches in this table view through the comparison between the queries RSSI, and the RSSI interval in the consistency view. The first matched row is returned, and all the others discarded. A response packet is created according with the returned row and forward back to the WSN.

Consistency views are a core resource in LIASensor, since it is the only resource that maps a query request to the responses size. By default, sensor nodes are deployed with a general consistency view. The default consistency view is not editable or erasable; however, the end-user can create and use, as he needs, new consistency views.

5 Evaluation

In order to evaluate and analyze the system prototype, we simulate an environment with 1 (one)

---

[1] http://castalia.research.nicta.com.au/index.php/en

[2] http://www.omnetpp.org

sensor per m^2 (square meter). The simulated environment has $1Km^2$ (one square kilometer) of total area. Thus, we simulate an environment with 1000 (one thousand) sensor nodes. The simulation was executed with one single end-user and the sensor nodes were deployed uniformly and randomly. We used elementary MAC and routing protocols, and the simulation was performed during a period equivalent to 10 hours of environmental monitoring which allow obtain promising results.

The results of the simulations were obtained through the comparison of three metrics: a) the number of transmitted messages, b) the size of the messages, and c) the lifetime of the entire network. These three criteria, allow understand the impact of LIASensor in a WSN. For each criterion, we made two simulations. In the first simulation we execute an elementary application protocol, whereas in the second simulation we execute a simulation with LIASensor as the application layer protocol. Both protocols used radius of interest made by one-hop transmission, i.e. only sensors that sensed the query were considered inside of the radius of interest. In addition, the end-user requested information every five minutes, making a 400 total requests.

- **Number of messages with Locality Awareness**

In order to analyze locality awareness, we simulate a network without interest awareness. Thus, the graph in figure 3 represents the results of a simulation executed with locality awareness turned on, and interest awareness turned off. In this graph we can confirm that locality awareness is capable of reducing the number of messages. We detect in average a decrease of 19% in messages number. This result can be justified with the consistency degrees that were formed throughout the network. As have been previously told, sensors positioned behind the last consistency degree do not respond, which results in a minor number of messages within the network.

- **Number of messages with Interest Awareness**

In the second simulation, we only test the impact of interest awareness over the network. Figure 4 shows the results of this simulation. So, we execute LIASensor with locality awareness turned off and interest awareness turned on. As expected, the number of messages decreased too. We notice a 37.8% decrease in average of messages numbers. These results are only possible, because interest awareness guarantees that only sensors with specifics skills answer the query.

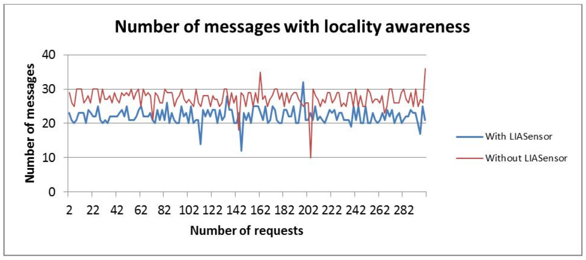

Fig. 3. Number of messages received in LIASensor with locality awareness.

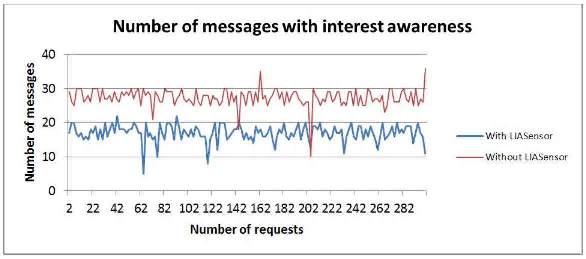

Fig. 4. Number of messages received in LIASensor with interest awareness.

- **Number of messages with Interest Awareness and Locality Awareness**

In the third simulation, we test LIASensor fully operational, i.e. with locality and interest awareness turned on. Figure (figure 5) presents the results. This graph clearly shows a decreasing in messages numbers. As explained before, locality awareness avoids messages from farther sensors, whereas interest awareness avoids useless messages to the user. Thus, as expected, when we combine these two techniques, we decrease, even more, the number of messages. With both interest and locality awareness we notice, in average, a decrease of 48% in messages numbers.

- **Size of messages with Interest Awareness and Locality Awareness**

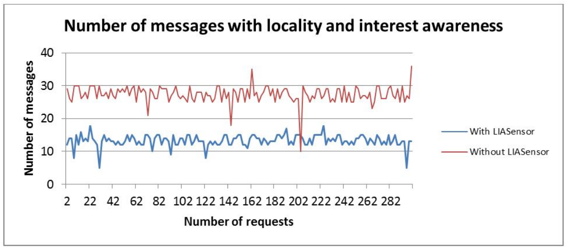

Fig. 5. Number of messages received in LIASensor with interest and locality awareness.

The number of messages is compared in the fourth simulation. Figure 6 shows the results of this simulation. The graph represents the average size of a packet containing environmental data.

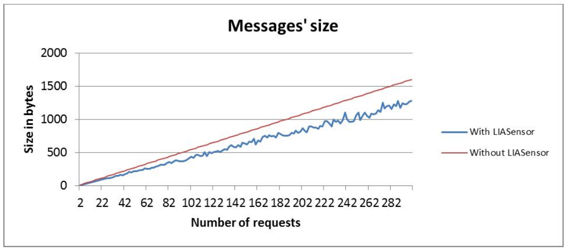

Fig. 6. Average packet size in LIASensor.

As expected, LIASensor reduces the size of messages within a WSN. Through the graph analyzes, we can infer that LIASensor can decrease packets size some minutes after the network deployment. LIASensor reduced the size of packets in 21.9%. This behavior is explained by the use of locality awareness, since this feature associates the sensor responses size with the sensor distance to the end-user.

- **Network Lifetime**

The last simulation tests the network lifetime. We execute this simulation until the death of the network. It was considered that a network was dead when all the sensors within the most internal consistency degree did not respond.

Table 2 shows the results of this simulation. The obtained results show a 16.4% of improvement. Without LIASensor, the network died after 15.26 hours of execution, whereas with LIASensor, the network only died after 18.26 hours of execution. These results are in accordance with the other results.

	With LIASensor	Without LIASensor
Time in hours	18.26	15.26

Table 2. Network lifetime with LIASensor.

As it is known, sensors spent a large amount of their energy when they are sensing and transmitting data, so by reducing the number of messages circulating in the network, as well as, the size of the messages in it, the increase of the network lifetime is a natural result.

6 Conclusion

All the results obtained were very encouraging. The simulations numbers shows that with LIASensor the network improve its lifetime. Moreover, the results proved that interest and locality awareness can achieve very good results. In fact, each one of these techniques can reduce the number of messages individually.

The explanation for these results can be easily extracted from LIASensor architecture. The architecture achieved to prioritize messages from sensors closer to the user rather than messages from sensors farther. In addition, the reduction of the amount of information within every response also contributed for these good results.

LIASensor is an application layer protocol, which allows the use of bottom layer API. This feature guarantees its use over any routing and MAC protocol without significant adaptations.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102 – 114, aug 2002.
2. J. Al-Karaki and A. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6 – 28, dec. 2004.
3. L. P. Clare, G. J. Pottie, and J. R. Agre. Self-organizing distributed sensor networks. pages 229–237, 1999.
4. M. Dong, K. Yung, and W. Kaiser. Low power signal processing architectures for network microsensors. In *Low Power Electronics and Design, 1997. Proceedings., 1997 International Symposium on*, pages 173 –177, aug. 1997.

5. P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June 2003.
6. L. A. Klein. *Sensor and data fusion: a tool for information assessment and decision making*, volume 324. SPIE press Bellinghamˆ eWA WA, 2004.
7. L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575 – 578, 2002.
8. L. Li and F. Bai. Analysis of data fusion in wireless sensor networks. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 2547 –2549, sept. 2011.
9. K. L. Morse. Interest management in large-scale distributed simulations, 1996.
10. N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, 2005.
11. G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, May 2000.
12. N. Santos, L. Veiga, and P. Ferreira. Vector-field consistency for ad-hoc gaming. In *Proceedings of the 8th ACM/IFIP/USENIX international conference on Middleware*, MIDDLEWARE2007, pages 80–100, Berlin, Heidelberg, 2007. Springer-Verlag.
13. J. Zheng and A. Jamalipour. *Wireless Sensor Networks: A Networking Perspective.* Wiley, 2009.