A Survey on Wireless Sensor Networks Deployment

ZORAN BOJKOVIC, BOJAN BAKMAZ
Faculty of Transport and Traffic Engineering
University of Belgrade
Vojvode Stepe 305, 11000 Belgrade
SERBIA
z.bojkovic@yahoo.com, b.bakmaz@sf.bg.ac.yu

Abstract: - In recent years extensive research has opened challenging issues for wireless sensor networks (WSNs) deployment. Among numerous challenges faced while designing architectures and protocols, maintaining connectivity and maximizing the network lifetime stand out as critical considerations. WSNs are formed by a large number of resource-constrained and inexpensive nodes, which has an impact on protocol design and network scalability. Sensor networks have enabled a range of applications where the objective is to observe an environment and collect information about the observed phenomena or events. This has lead to the emergence of a new generation sensor networks called sensor actuator networks. Approaches developed to query sensor-actuator networks (SANETs) are either application-specific or generic. Application-specific SANETs provide limited reusability, are net cost effective and may require extensive programming efforts to make the network able to serve new applications. A WSNs should be able to operate for long time with little or no external management. The sensor nodes must be able to configurate themselves in the presence of adverse situations. In this work, dealing with challenges for WSNs deployment, we start with mobility-based communication in WSNs. Then, we introduce service-oriented SANETs (SOSANETs) as an approach to build customizable SANETs. In the second part, we describe localization systems and analyze self configurability, situation awareness and intrusion detection system. In the third part, we present wireless distributed detection as well as a model for WSN simulation. Finally, conclusions and proposals for future research are given.

Key-Words: Mobility-based communication, sensor-actuator networks (SANETs), service oriented SANETs (SOSANETs), wireless sensor networks (WSNs).

1 Introduction

In recent years extensive research has opened challenging issues about performance evaluation for wireless sensor networks (WSNs). These promising technologie can be used to achieve a variety of goals, from health monitoring to industrial automation, emergency management and environmental monitoring [1]. They are designed to collect data and report to a central unit, connected to the Internet or monitored.

Physical layer issues are essential to the success and effectiveness of any wireless technology. Spread spectrum is a technique that has been used successfully in a many contexts, including both cellular communication and data communications in the industrial, scientific and medical (ISM) band. For example, various 802.11 (Wi-Fi) standards and 802.15.4 (ZigBee) radios all use spread spectrum in the ISM band [2]. Sensor nets face technical problems similar to those of mobile ad hoc networks (MANETs). WSNs are formed by a large number of resource-constrained and inexpensive nodes, which has an impact on protocol design and network scalability. Energy is a primary concern, because nodes usually run on nonrechargeble batteries. Thus, the improvement of network lifetime is one of a fundamental research issue [3]-[5]. The applications for WSNs and generation of the protocol layers are driven by physical sensor measurements, rather than voice or user-data services [6]. Sensor networks have enabled a range of applications where the objective is to observe an environment and collect information about the observed phenomena or events. In many cases appropriate actions must be taken upon the occurrence of a given event. This has led to the emergence of a new generation of sensor networks called sensor actuator networks (SANETs) that have sensor nodes and actuator nodes.

In order to monitor an area of interest a large number of sensor nodes cooperate among themselves. Several physical properties can be monitored by a WSN (temperature, humidity, pressure, ambient light and movement). The collected information and sensor nodes must be localized in space to identify the location of an event. This positioning is accomplished using a localization system. These systems are key part of WSNs. They not only locate events, but can also be used as the

base for routing, density control, tracking, and number of other protocols. Due to other key role in WSNs, localization systems can be a target of an attack [7].

Sensor can be fully autonomous due to their battery-powered computational and communication capabilities. As a result, a sensor network should work without any human assistance during most of its lifetime. As a requirement to be self-configurable, a sensor node must build on situation awareness mechanisms, capable of detecting the presence of unusual events, without consuming many of its resources. Thus, these mechanisms can serve as a foundation for more complex schemes, such as an intrusion detection system (IDS) [8].

Sensors in WSNs detect environmental variations and then transmit the detection result to a fusion center. The fusion center collects all detection results and determines the phenomenon that is denoted by local decision which is made by the sensor [9]. Wang, et al. [10] proposed Distributed Classification Fusion using Error-Correcting Codes (DCFECC) for fault-tolerance by combining the distribution detection theory [11] with the conception of ECC in communication systems [12]. One sample is detected in each of N sensors for a given phenomenon. A codeword consisting of N symbols is designed for each phenomenon. A one-dimensional code (1 x N) corresponds to a phenomenon. Thus, M phenomena form an M x N code matrix. Each symbol with one bit is assigned to each sensor. A local decision is made from the detection results and is represented with the assigned symbol. Binary decisions from local sensors, possibility in the presence of faults, are forwarded to the fusion center that determines the final decision. Each codeword in the code matrix is chosen apart from each other and can tolerate faults made on local decisions when making the final decision. This approach not only provides an improved fault-tolerance capability but also reduces computation time and memory requirements at the fusion center. Distributed classification fusion using soft decision decoding (DCSD) was developed by improving DCFECC. DCSD adopts a symbol with L bits, instead of one bit [13]. However, the misclassification probability remains high in the extreme case (many faulty sensor and very low signal-to-noise ratios). Moreover, the multi-bit symbol increases the sensor complexity - cost.

2 Mobility-based Communication in Wireless Sensor Networks

A large number of small and simple sensor devices communicate over short-range wireless interfaces to deliver observations over multiple hops to central locations called sinks. Sensor nodes, and hence these applications, are subject to constraints such as limited processing, storage, communications capabilities and limited power supplies. Numerous challenges are faced while designing WSNs and protocols, maintaining connectivity and maximizing the network lifetime over critical considerations. The connectivity is met by deploying a sufficient number of sensors, or using nodes with long-range communication capabilities to maintain a connected graph. The network lifetime is directly related to how long the power services in sensor nodes will last. The network lifetime can be increased through energy conserving methods such as using energy-efficient protocols and algorithms, and battery replenishment techniques.

The mobile devices can also be used as an orthogonal method to address the connectivity and lifetime problems in WSNs. In other scenarios mobile devices can be incorporated into the design of the WSN architecture such as airborne and ground-based vehicles. With communication devices on mobile platforms, the connectivity and energy efficiency (network lifetime) problem are addressed as follows: As for connectivity, mobile platforms can be used to carry information between isolated parts of WSNs. On the other hand, energy efficiency means that information carried in mobile devices can reduce the energy consummation of sensor nodes by reducing multihop communication. In recent years a number of approaches exploiting mobility for data collection in WSNs have been proposed. These approaches can be categorized with respect to the properties of sink mobility and the wireless communications methods for data transfer.

Mobile base station (MBS) is a mobile sink that changes its position during operation time. Data generated by sensor are relayed to MBS without long term buffering.

Mobile data collector (MDC) is a mobile sink that visits sensors. Data re buffered as source sensor until the MDC visits the sensors and downloads the information over a single-hop wireless transmission.

Rendezvous-based solutions are hybrid solutions where sensor data is sent to rendezvous point close to the path of mobile devices. Data are buffered at rendezvous point until they are downloaded by mobile devices.

Comparison of mobility-based communication proposals is given in Table 1.

Table 1. Comparison of mobility-based communications proposals

Class	Mobile base station	Mobile data collector	Randezvouz
Multihop commun.	Yes	No	Yes
Long-term buffering	No	Yes	Yes
Mobility	Controlled	Random, predictable, controlled	Controlled
Message latency	Low	High, medium	Medium
Platform mobility	Low to very high	High, medium	Medium, high
Energy consuption	High	Low	Medium, low

3 Sensor-Actuator Networks

Sensor-actuator networks (SANTEs) have sensor nodes and actuator nodes. Sensor and actuators communicate and collaborate to perform distributed sensing and acting tasks. Sensors gather information about the physical world, while actuators make decisions and perform actions that affect the environment. Actuators are able to change parameters in their environment. Applications of SANETs include environmental applications business applications, health applications, home automation, and entertainment. In recent years the need to decouple SANET's has led to the emergence of generic SANETs, an alternative design model where an application - independent query system is developed on the SANET. In this model, the query system is designed to answer queries from any application [14].

In application specific SANET deployments, the application consists of a distributed code installed on some or all of the nodes of the network. In simple applications, the some code is installed on all nodes. In more complex applications, different code modules are installed on different nodes. Generic SANETs are not intended to be used by a specific application. They usually require that a generic code (i.e., the query processing system) to be installed on all nodes of the network. Current design models for sensor - actuator systems seem increasingly unable to cope with the requirements of the next generation of open, ubiquitous interoperable, multipurpose SANETs. Architectures for future sensor systems will have to be able to serve different applications and adopt to different post deployment query patterns. To enable next-generation sensor - actuator systems, new customizable architectures are needed.

Customizable SANETs are readily configurable to serve different types of applications with arbitrary query patterns. Customizable SANETs would provide developers the flexibility to combine the resources provided by nodes in one or more SANETs to meet the requirements of new applications and yet expect the some level of performance that would result from an application - specific deployment. A possible alternative to building customizable SANETs is to use generic SANETs as their backbone and develop additional software layers that customize the functionalities of generic SANETs to satisfy the requirements of the given application. This leads to further lower performance and memory availability.

Service - oriented SANETs, or SOSANETs is a novel approach to building customizable SANETs. In SOSANETs, nodes reusing and actuation capabilities are exposed to applications in the form of a collection of programmatic abstractions called services. A service deployed on a node is a lightweight code that provides some functionality supported by the node. These services may be individually invoked or combined in complex ways to form a virtual SANET with far reacher sensing and actuation capabilities. Services are deployed directly on a top of the operating systems, and they are accessible directly by applications.

In service - oriented query model for SANETs, nodes have heterogeneous sensing and actuation capabilities. Each node exposes its capabilities as services. A service is a computational component that:

a) has a unique network-wide identifier,
b) may be invoked asynchronously,
c) may have one or more parameters,
d) produces one or more values as a result of invocation.

The SOSANET has one or more base stations. Users invocate the SOSANET by submitting queries to one of its base stations or directly to individual nodes. The software running on top of the operating

system at each node is organized into three layers: service - oriented query layer, service layer and routing layer. An overview of a node's architecture that supports the service - oriented query model is shown in Fig. 1.

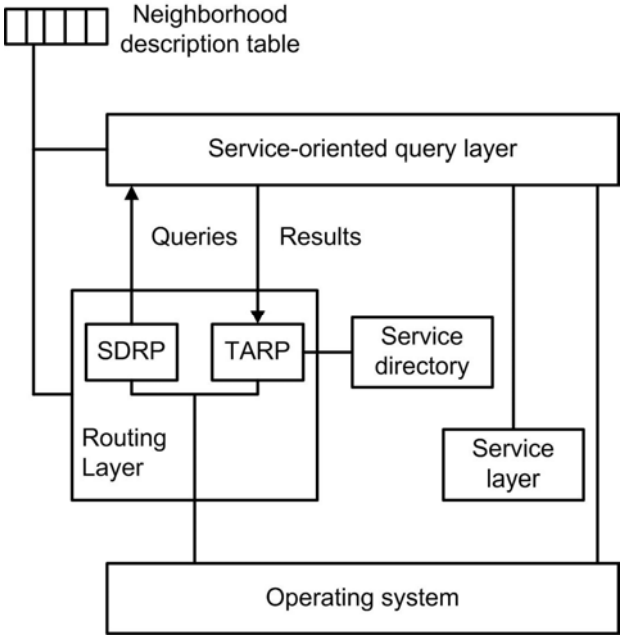

Fig. 1. Node's architecture in service - oriented networks

Service - oriented query (SOQ) layer receives queries from the service - driven routing layer interprets them, invokes the appropriate services specified in the queries, collects the results from the services packages these results into query result messages, and submits those messages to the service - driven routing layer to send them to the query issues. As shown in Fig. 2, the SOQ layer consists of two main modules: service invocation scheduling module (SISM) and event detection module (EDM).

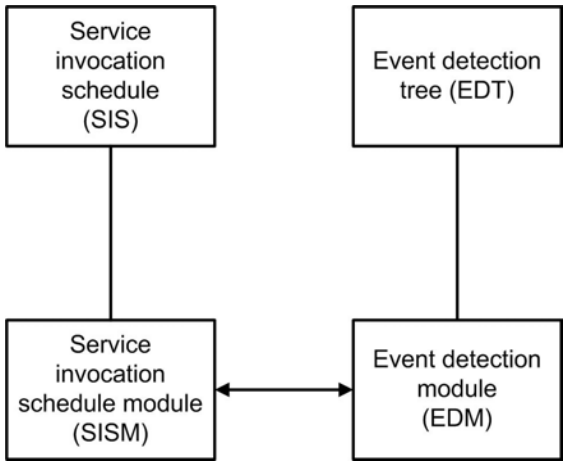

Fig. 2. Service - oriented query layer

Service invocation scheduling module (SISM) monitors the node's query load and schedules service invocation while considering the frequency and expiration time of the different queries. The SISM maintains a list of services to be invoked and the times of invocation in a service invocation schedule (SIS). This module also conducts multiquery optimization by exploiting any relationships that may exist between several queries.

Even detection module (EDM) detects the events that are relevant to the current query load at the local node. The EDM maintains an event list of all the events relevant to the current query load. It also maintains a mapping between attributes and events. Each time a change is detected in the value of an attribute *a*, the EDM evaluates the event predicates whose value depends on the value of the attribute *a*. The EDM then activates all the queries whose event clause events to time.

The service layer is a collection of light weight services. Each service is a software module that carries out some sensing, actuation, or control function. A service may interact directly with oriented service components (sensor controller timers) of its local node. These components interact with the node's hardware modules, like actuation unit, clock, etc.

The routing layer is used for delivering incoming queries to the service-oriented query layer (SOQ) layer on the local node, sending out query results produced by the SOQ layer and for forwarding received queries and query results to neighbors. This layer consists of two protocols: service-driven routing protocol (SDRP) and trust aware routing protocol (TARP). SDRP routers queries from the base station to the nodes in the network, while TARP routers query results from the network's nodes to the base station.

4 Localization Systems

To be deployed in hostile environments, WSNs require a secure localization system, in which we must solve the localization problem but also must be aware that we are in the presence of compromised nodes - malicious nodes or network nodes that have been corrupted by a malicious code - and/or a compromised environment - where hostiles can change the characteristics of an environment and also have physical access to nodes.

From the view point of localization systems, we have two types of nodes: regular nodes and beacons. Regular nodes refer to nodes in the network that have no knowledge of their position and no special hardware to acquire this information. Beacon nodes

also known as land-marks or locators are nodes that do not require a localization system to estimate their physical positions. In fact, they form the base of these systems. Their position is obtained by manual placement or external means such as a global positioning system (GPS). Distributed node localization algorithm named mobile beacons–improved particle filter (MB-IPF) was proposed in [15]. In the algorithm, the mobile nodes equipped with GPS move around WSN field based on the Gauss-Markov mobility model and periodically broadcast the beacon messages. Each unknown node estimates its location in a fully distributed mode based on the received mobile beacons.

In a localization system, the problem obtain arises is: given a multihop network and a set of beacon nodes with their known positions, we have to find the position (for example latitude, longitude) of regular nodes based on available information. Localization systems can be divided into three distinct components shown in Fig. 3.

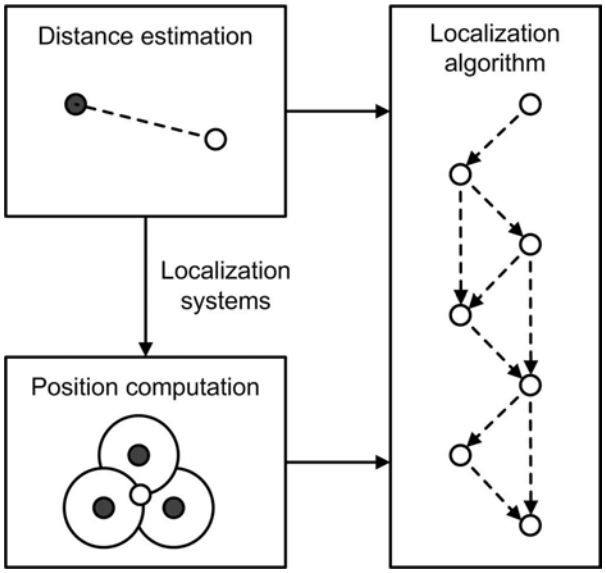

Fig.3. Distinct components in the division of localization systems

Distance estimation component is responsible for estimating information regarding the distance and/or angles between two nodes. Recognized techniques used in this component include received signal strength indicator, time of arrival, number of hops, or angle of arrival.

Position computation component is responsible for computing the position of a node based on available information about the distance/angles and position of represent nodes. Some techniques used to compute a position include trilateration, multilateration, or triangulation.

Localization algorithm is the main component of a localization system. It determines how the available information will be manipulated to enable most or all of the nodes of the WSN to estimate their positions. It is a distributed and usually multihop algorithm. Some known algorithm include the Ad hoc Positioning System (APS) as well as Direct Position Estimation (DPE).

5 Self-configurability, Situation Awareness and Intrusion Detection System

A specific feature of sensor nodes is their inherent autonomy. By means of their computational capabilities, nodes can analyze the data coming from their embedded sensing units. Additionally, they operate without any pre-existing infrastructure because they can communicate with their surroundings using wireless transceivers. Also, they can survive in certain configuration, because they are powered by small batteries. Due to this autonomy, sensor nodes should behave as self-configurable entities [8]. Also similar to ad hoc networks, WSNs will most likely be required to self-configure into connected networks, but the difference in traffic, energy trade-offs and other issues could require new solutions. This includes the need for sensor nodes to learn about their location.

In order to be fully autonomous and self-capable, it is fundamental for the nodes to be aware of their environment. It is important to recognize certain events that might affect the behavior of the network. For example, nodes that are affected when one of the routers of network fails must be able to notice automatically and react. The task of detecting such events relies upon the existence of the situation awareness mechanisms. Without these mechanisms, a node can not understand fully the current position of its environment and will not be able to configure itself to respond to internal/external events. These mechanisms must be sufficiently effective, flexible and simple in the same time to enable their execution in the constrained nodes. To fulfill these requirements, the properly detection of the problematic events that can occur in a sensor network must be defined as a crucial factor.

There are some mechanisms that try to detect abnormal situations caused by malicious nodes, either by analyzing the behavior of the network, or by using protocol-specific technologies such as for example, automate theory. An intrusion detection system (IDS) is an interesting, underdeveloped service, useful for scenarios where there is a

possibility for a node being subverted and controlled by an adversary. The major task of an IDS is to monitor computer networks and systems to detect eventual intrusions in the network, alert users after specific intrusions have been detected and finally, if possible reconfigure the network and mark the root of the problem as malicious [8]. Aside from the detection of abnormal events, there are other aspects in the development of an IDS that must be solved (for example the exact location of the detection agents and their tasks). On the other hand, when considering the existence of a fully functional IDS, there is a need for filtering the information provided by the system to detect malicious nodes and distinguish between possible errors and attacks launched against the network.

There are two main types of approaches to intrusion detection [16]:

- Misuse (signature-based) detection, where known security attack signatures are kept and matched against the monitored system. This type of detection can accurately detect known attacks, but it is unable to detect any new attacks that emerge in the system.
- Anomaly detection, where a normal profile of the monitored data is established, and then anomalies are identified as measurements that deviate from default profiles. Because of that, anomaly detection is capable of detecting new types of security risks. A problem with this approach is the high level of false alarms. Due to, reducing level of false alarms while still being responsive to detecting security risks, is major issue for intrusion detection.

Functional IDS have to fulfill multiple objectives related to accurate intrusion detection using various ingredients like:

- Intrusion checkpoints represent the observable states of the IDS and analyze the sensor activity that predicts the transition form normal to intrusion state.
- Creation of an activity profile that identifies abnormal activity of the observable states by measuring the sensor deviation from normal behavior.
- Concept drift that measures the change in user behavior over a period of time.
- Control loop which adopts the trigger based on the weighted sum of proportional, average, and derivative sensor measurements over derivative and integral time window.
- Model that predicts the most probable state based on previous state as well as observed states. This can be accomplished using hidden Markov model described in [17].

6 Wireless Sensor Network for Distributed Detection

Distributed detection of certain events in the environment is an important application of sensor networks. The traditional approach of studying the distributed detection problem is to assume that sensors transmit their observation through a parallel access channel, which is independent across sensors. For large-scale sensor networks, this assumption implies a large bandwidth requirement for simultaneous transmissions or a large detection delay. Alternatively, we can employ a multiple access channel, whose bandwidth requirement does not depend on the number of sensors, but due to the additive nature of the channel, the received signal at the fusion center is generally not sufficient for reliable detection [18].

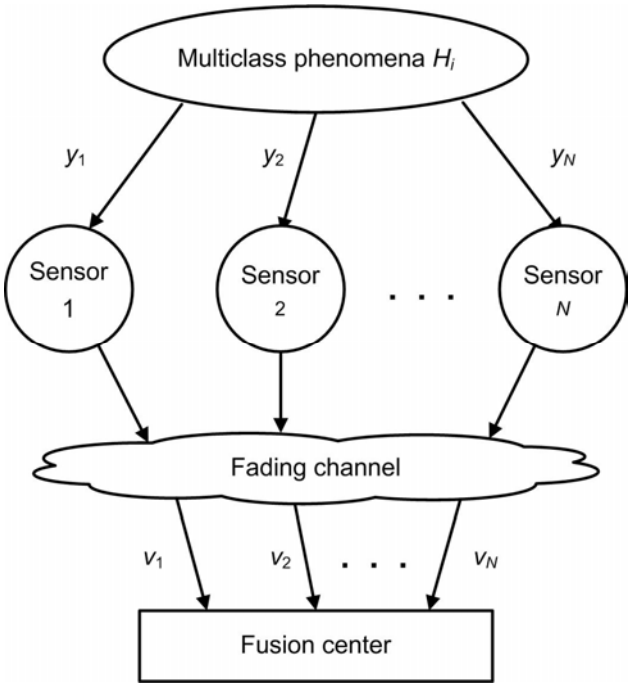

Fig. 4. Wireless sensor network for distributed detection using N sensors

A wireless sensor network for distributed detection with N sensors deployed for collecting environment variation data and a fusion center, for making a final decision of detections is depicted in Fig. 4. When one of the phenomena H_i (i = 1, 2, … , M) occurs, all sensors observe the same phenomena.

One observation y, is undertaken at the j-th sensor. The observation is normally a real number represented by infinite number of bits. Transmitting the real number to the fusion center would consume too much power, so a local decision, n_j, is mode instead. For a phenomenon, if only L bits are allowed to send the local decision from the sensor to the fusion center, then the L bits are used to represent the decision. The fusion center collects all local decisions and makes a global decision according to them.

In [13], a person-by-person optimization is adopted to determine all of the local decision rules. The decision region at sensor j can be represented by a set of thresholds such that a local decision rule associated with this threshold set can be performed to determine n_j when y_i is observed.

7 Updating Software in Wireless Sensor Networks

A critical issue in the effective deployment of WSNs is the ability to update software after deployment. There are a number of reasons why the software may require updating. The Software Engineering Institute (SEI) at Carnegie-Mellon University identifies four categories of software updates for defendable systems, which help to provide an insight into these reasons: maintenance releases, minor releases, major releases (technology refresh), and technology insertion. Embedded wireless sensor systems programmed by specialists are likely to experience higher levels of maintenance than normal. Minor release will be used to improve data collection and performance. As the needs of WSNs are likely to develop dynamically over time, major releases can be expected in response. Finally, due to the active research on WSNs related technologies and the associated development of new algorithms as well as protocols technology implementation, all this will be an important driver of software updates [19].

Wireless sensor nodes are characterized by very limited resources and by large-scale deployment. Accessing these nodes in the field to perform software updates can be difficult to locate or inaccessible, or the scale of the deployment can preclude individual access. Remote update poses its own problems. Three key issues are:

- Avoiding interference with data collection while sharing the same communication infrastructure;
- Minimizing the cost of upgrades in terms of the impact on sensor network lifetime;
- Avoiding the loss of part or all of a sensor network due to an upgrade fault.

General software update model for WSNs is shown in Fig. 5. The high level data – flow diagram highlights the interactions between the three key elements of software update functionality: generation, propagation and activation.

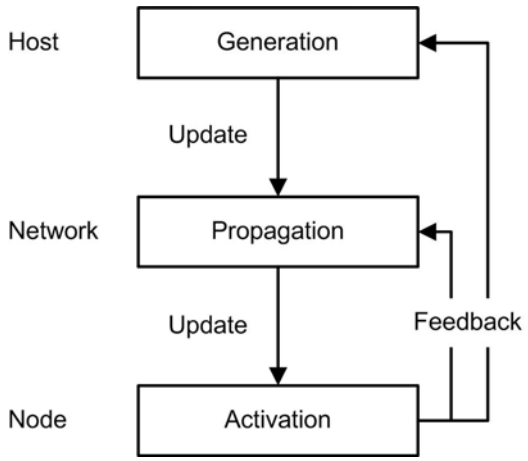

Fig. 5. General software update model for WSNs

On a host system, generation functionality is connected with planning for the update, generating the update data, and inserting it into the network. Propagation functionality is connected with transferring the update from the insertion point through the WSN to desired destination points or targets. This network-wide functionality is supported by client-server interactions. Activation functionality is associated with initiating the execution of the software update on the destination nodes. It may be triggered locally, through consensus, or from the host node, based on various rules.

8 Model for Wireless Sensor Network Simulation

WSNs are composed of a large number of sensors or nodes, which gather events and process them. Some WSNs and simulation tools also include sink nodes. They process data from the net and may interrogate sensors about events of interest. The events come from the physical environment component, which may be generated by itself, or triggered by agents.

Due to the hard constraints of sensors, the classical layered approach is not suitable. Node behavior depends on interacting factors that cause cross-layer interdependences. A convenient way to describe it is to divide nodes into tiers as represented in Fig. 6 [3].

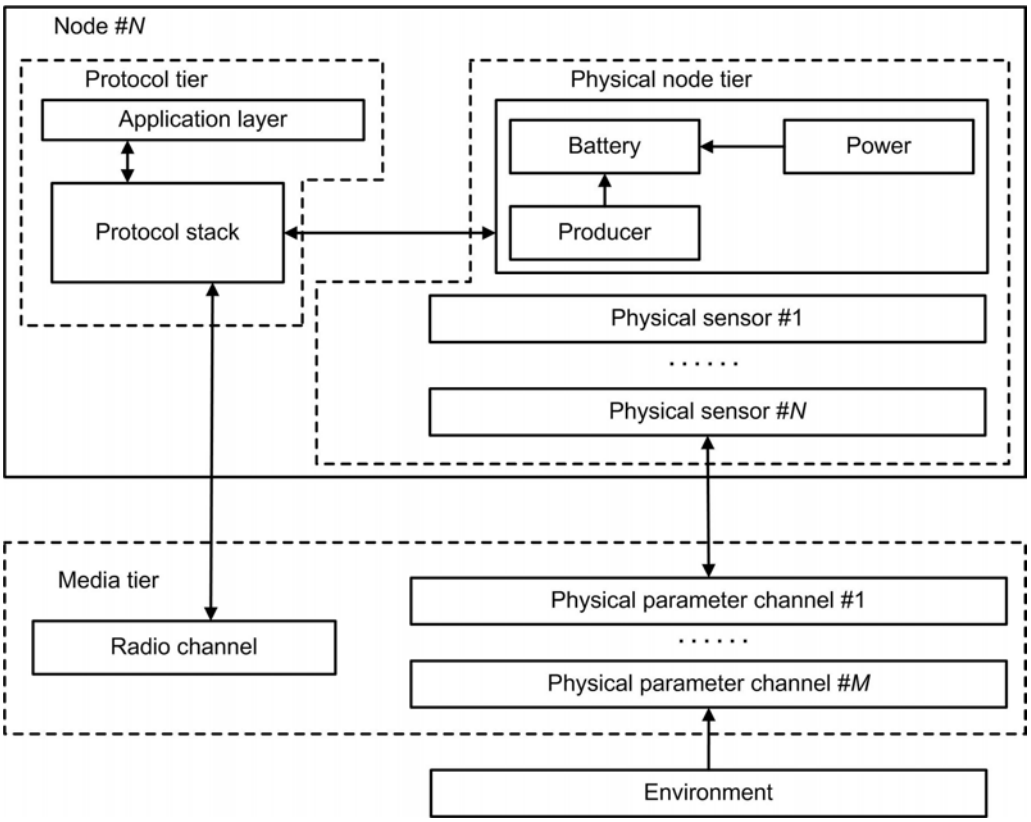

Fig. 6. Network-level model for wireless sensor node

The protocol tier comprises all the communication protocols. Their operation usually depends on the state of the physical tier. The physical node tier represents the underlying hardware and measurement devices. The media tier links the node to the real world through a radio channel and one or more physical channels, connected to the environment component.

Architecture layers need to exchange information that would be isolated in the open system interconnection (OSI) model. This tight coupling affects the simulation architecture in different ways. It is important to pointout that the overall design must provide an efficient mechanism to share information between modules, without degrading performance. Second, the interface between components must be flexible and extensible. Fixed interfaces and primitives between layers or components should not be assumed. For example, an estimate of link state may be used by medium access control (MAC), routing and application layers.

Consumption should be controlled by means of two different modules: the power module and the battery module. The power module computes the power consumption of the different components, while the battery module uses this information to compute the battery discharge. Besides, energy producers inside the nodes may be considered, for example, to model solar or even wind-powered sensors. These components are introduced via producer modules connected to the battery module.

Synthesized radio channel must determine the nodes that receive a transmission, the quality of reception (with or without error) as well as the state of the shared medium (busy or free). To implement such functionality, simulators employ three independent modules.

The transmission module defines the radiated power, frequency, data rate and other transmission parameters. The propagation module computes the received power which is mainly a function of the transmission parameters and distance. The propagation model used can be deterministic (e.g., free space, two-ray ground reflection) or add some random component (e.g., shadowing). The reception module decides whether packets are received, whether there is an error, or whether the medium is busy.

In the structure of the environment model, sensors are fed with data from the environment through physical channels. These channels are in charge of deciding when and which nodes receive the physical events generated by the physical event generator. Some WSN simulators incorporate independent agents (for example, mobile vehicle) that trigger events in the physical event generator.

9 Conclusions and Future Research

This article surveys wireless sensor networks deployment. We describe mobility-based communication in WSNs. The emphasis is on sensor actuator networks (SANETs), node's architecture in service oriented SANETs (SOSANATs) as well as service oriented query layer. Then, we describe intrusion detection system and show that a sensor network must be defined as a crucial factor. Updating software in WSNs together with model for simulation is presented, too.

Mobility based communication can prolong the lifetime of WSNs and increase the connectivity of sensor nodes and clusters. In contrast with current SANETs, SOSANETs expose their sensing, and actuation capabilities in the form of service that may be invoked by any application. The potential of SOSANETs can be shown in addressing the limitations of current SANET architectures. An increase localization system can be attacked in a number of ways to compromise the entire functioning of a WSN and thus lead to incorrect military plans and decision making. We divide localization systems into three different components: distance/angle estimation, position computation and localization algorithm. Using its embedded sensors and the wireless channel, a sensor node can feel and interact with the world that surrounds it. However, there is a difference between feeling the world and understanding the world. It is possible to reduce this gap using certain situations awareness mechanisms.

Among the open research problems, real-time solutions that result in low mobile device speeds and cooperation between multiple mobile devices stand out as challenges that have significant impact. The adaptation of solutions to WSNs with dynamic requirements should also be investigated as near-term research directions. Future SANETs will require new architectures. We forsee service - oriented architectures as a highly viable candidate to support the requirements of tomorrow's sensor networking.

References:

[1] A. Mitseva et al., "CRUISE research activities toward ubiquitous intelligent sensing environments", *IEEE Wireless Communications*, vol. 15, no. 4, Aug. 2008, pp. 52-59.

[2] J. G. Andrews, S. Weber, M. Haenggi, "Ad hoc networks: to spreed or not to spread?", *IEEE Communication Magazine*, vol. 45, no. 12, pp. 84-91, Dec. 2007.

[3] E. Egea-Lopez, et al., "Simulation scalability issues in wireless sensor networks", *IEEE Communication Magazine*, vol. 44, no. 7, pp. 64-73, July 2006.

[4] B. Hegyi, J. Levandovszky, "Optimal Statistical Energy Balancing Protocols for Wireless Sensor Networks", *WSEAS TRANSACTIONS on COMMUNICATIONS*, vol. 6, no. 5, May 2007, pp. 689-695.

[5] J. Levendovszky, et. al., "Energy balancing by combinatorial optimization for wireless sensor networks", *WSEAS TRANSACTIONS on COMMUNICATIONS*, vol. 7, no. 2, Feb. 2008, pp. 27-32.

[6] I. F. Akyildiz, X. Wang and W. Wang, "Wireless mesh networks: a survey", *Computer Networks*, vol. 47, no. 5, pp. 445-487, Mar. 2005.

[7] A. Boukerche, et al., "Secure localization algorithms for wireless sensor networks", *IEEE Communication Magazine*, vol. 46, no. 4, pp. 96-101, Apr. 2008.

[8] R. Ramen, J. Lopez, S. Gritzalis, "Situation awareness mechanisms for wireless sensor networks ", *IEEE Communication Magazine*, vol. 46, no. 4, pp. 102-107, Apr. 2008.

[9] H-T. Pai, Y. S. Han, J-T. Sung, "Two dimensional coded classification schemes in wireless sensor networks", *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1450-1455, May 2008.

[10] T. Y. Wang, et al., "Distributed fault - tolerant classification in wireless sensor networks", *IEEE Journal on Selected Areas Communications*, vol. 23, no. 4, pp. 724-734, Apr. 2005.

[11] P. K. Varshney, *Distributed Detection and Data Fusion*, New York, Springer, 1997.

[12] F. J. Mac Williams, N. J. A. Sloane, *The Theory of Error Correcting Codes*, New York, Elsevier, 1977.

[13] T. Y. Wang, et al., "A combined decision fusion and channel coding scheme for distributed fault-tolerant classification in wireless sensor networks", *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1695-1705, July 2006.

[14] A. Kezgni, M. Eitoweissy, "Service-oriented sensor-actuator networks", *IEEE Communication Magazine*, vol. 45, no. 12, pp. 92-100, Dec. 2007.

[15] K. Xing-Hong, S. Hui-He, "Localization Assisted by the Mobile Nodes in Wireless Sensor Networks" *WSEAS TRANSACTIONS on COMMUNICATIONS*, vol. 6, no. 8, Aug. 2007, pp. 767-772.

[16] S. Rajasegarar, C. Leckie, and M. Palansiwami, "Anomaly detection in wireless sensor networks", *IEEE Wireless Communications*, vol. 15, no. 4, Aug. 2008, pp. 34-40.

[17] R. Khanna, H. Liu, "Control theoretic approach to intrusion detection using a distributed hidden Markov model", *IEEE Wireless Communications*, vol. 15, no. 4, Aug. 2008, pp. 24-33.

[18] W. Li, H. Dai, "Distributed detection in wireless sensor networks using a multiple access channel", *IEEE Transactions on Signal Processing*, vol. 55, no. 3, Mar. 2007, pp. 822-833.

[19] J. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale", *Proc. 2nd Int. Conf. Embedded Network Sensor Systems*, Baltimore, MD, Nov. 2004, pp. 81-94.

[20] Z. Bojkovic, B. Bakmaz, M. Bakmaz, "Some security trends over wireless sensor networks", *Proc. 12th WSEAS International Conference on COMMUNICATIONS*, Heraklion, Greece, Jul. 2008, pp. 470-474.