International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:7, 2010

Position Awareness Mechanisms for Wireless Sensor Networks

Seyed Mostafa Torabi

***Abstract*—**A Wireless sensor network (WSN) consists of a set of battery-powered nodes, which collaborate to perform sensing tasks in a given environment. Each node in WSN should be capable to act for long periods of time with scrimpy or no external management. One requirement for this independent is: in the presence of adverse positions, the sensor nodes must be capable to configure themselves. Hence, the nodes for determine the existence of unusual events in their surroundings should make use of position awareness mechanisms. This work approaches the problem by considering the possible unusual events as diseases, thus making it possible to diagnose them through their symptoms, namely, their side effects. Considering these awareness mechanisms as a foundation for high-level monitoring services, this paper also shows how these mechanisms are included in the primal plan of an intrusion detection system.

***Keywords*—**Awareness Mechanism, Intrusion Detection, Independent, Wireless Sensor Network

I. INTRODUCTION

A Wireless sensor network (WSN) consists of a set of battery-powered nodes, which collaborate to perform sensing tasks in a given environment. It may contain one or more base stations to collect sensed data and possibly relay it to a central processing and storage system. The communication range of individual sensor nodes is generally limited, and communication is often carried out in a multi-hop manner. The main purpose of a WSN is to serve as the bridge between the real world and a computer system, providing physical information such as temperature, light, and radiation. Measuring the physical information relies on the tiny and highly constrained sensor nodes. A typical sensor network deployment can comprise from dozens to thousands of nodes that in a distributed way collect and send the information to a central device, the base station. This allows access to the services provided by the sensor network for any user of the computer system. All data coming from the nodes, as well as control commands directed to them, will traverse the base station.

Wireless Sensor nodes can be fully independent due to their battery-powered computational and communication capabilities. As a result of this independence, a sensor network should work without any human assistance during most of its lifetime. However, as a requirement for being self-configurable, a sensor node must build on position awareness mechanisms, capable of detecting the presence of unusual events, without consuming many of its resources.

Author is with the Islamic Azad University - Rafsanjan Branch
Rafsanjan, Iran. e-mail:torabi@iaurafsanjan.ac.ir

In fact, these mechanisms can serve as a foundation for more complex schemes, such as an intrusion detection system (IDS). IDSs are particularly useful in scenarios where there is the possibility that a node might be controlled by a malicious adversary.

This research elaborates on the importance of mechanisms for detecting unusual positions in sensor nodes, reviews the main research activities in this area, and presents a novel approach for the detection of events. This approach considers a static WSN as a living body; an unusual position as a disease; and associated with any disease, a set of symptoms that can lead to its diagnosis. By analyzing both diseases and symptoms, it is possible to develop lightweight awareness mechanisms. Additionally, we highlight how it is possible to integrate those procedures into an IDS architecture.

II. POSITION AWARENESS AND SELF-CONFIGURABILITY

A specific feature of sensor nodes is their inherent independence. 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. Furthermore, they can survive in their deployment site, even for years in certain configurations, because they are powered by small batteries. Due to this independence, sensor nodes should behave as self-configurable entities. They should be set up and deployed without major effort by non-experts, and they should be able to adapt and heal themselves during the lifetime of the network.

However, to be fully independent and self-capable, it is essential for the nodes to be aware of their environment, that is, to recognize certain events that might affect the behavior of the network. For example, nodes that are affected when one of the routers of the network fails to work must be able to notice automatically and react accordingly (Fig. 1). The task of detecting such events relies upon the existence of position awareness mechanisms. Without these mechanisms, a node cannot understand fully the current position of its environment and will not be able to configure itself to respond to internal/external events. Note that these mechanisms must be sufficiently lightweight to enable their execution in the constrained nodes.

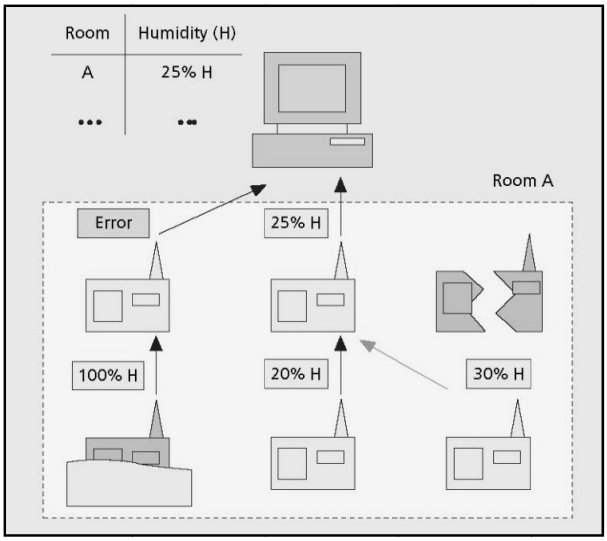

Fig. 1 Importance of self-awareness mechanisms for sensor networks

There are some existing techniques that enable the control of simple factors such as the actual position of the sensor nodes. For example, the protocol [1] consists of simply sending periodical "heartbeat" messages to other nodes to check whether they are alive. This technique was improved in [2] by sending that information to the base station while trying to minimize the use of resources. There are also other mechanisms that try to detect unusual positions caused by malicious nodes, either by analyzing the behavior of the network [3] or by using protocol-specific techniques such as automata theory [4].

These mechanisms also serve as a foundation for creating complex schemes like IDSs. An IDS is an interesting, albeit underdeveloped service, useful for scenarios where there is the possibility of a node being subverted and controlled by an adversary. The major task of an IDS is to monitor computer networks and systems to detect these 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. A standard and full-fledged IDS for sensor networks has not been defined yet, although some authors have explored how to develop mechanisms for it.

Aside from the detection of unusual events, there also are other aspects in the development of an IDS that must be solved. The exact location of the detection agents and their tasks is an example. In hierarchical configurations — where more powerful devices called cluster heads manage an entire cluster of nodes — full-fledged agents can be located at those powerful devices [5]. However, in flat configurations, the optimal distribution of the tasks through all the agents requires research. The redundancy of the network can be used as an advantage in this type of configuration, because (as detailed by the only major contribution in [6]) it can be possible to activate the detection tasks only in some nodes. 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 [7].

III. DEVELOPMENT OF LIGHTWEIGHT AWARENESS MECHANISMS

As mentioned previously, one of the key factors for the development of lightweight detection mechanisms is the knowledge of the problematic events that can occur in a sensor network, and how to properly detect them. For this purpose, it is possible to use the simile of "a sensor network as a living body," where a sensor node is considered the "cell" of the system, and the base station is the "brain," as seen in Fig. 2. Having in mind this simile, it is possible to consider that the presence of certain symptoms (i.e., collateral effects) will be indicative of the existence of a disease (i.e., an unusual event). Therefore, the detection mechanisms will infer the existence of unusual events based on the existence of their collateral effects.

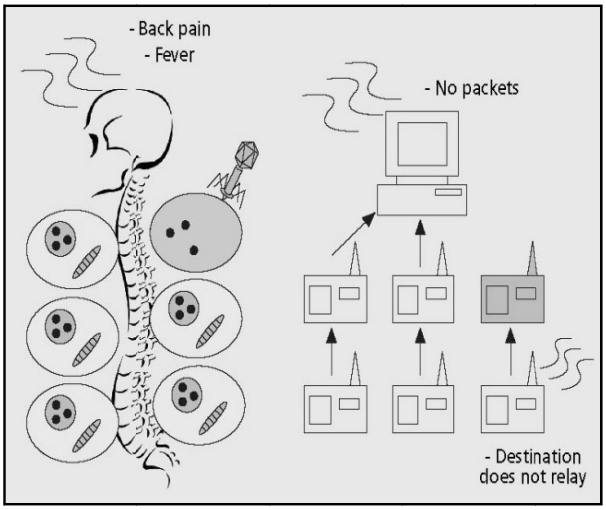

Fig. 2 A sensor network as a living being

One of the difficulties associated with the diagnosis of a disease consists in separating the existing symptoms from the normal behavior of the body. However, the functionality of a sensor network is usually fixed, with sensor nodes providing the same services during the entire lifetime of the network. Therefore, any deviation of the behavioral pattern of the network, or the existence of a well-established set of unusual patterns, can be considered to be a potential effect of an unusual event. Another issue that can affect the diagnosis is to distinguish one disease from another, given the existing symptoms. Nevertheless, in a sensor network context, the mere possibility of detecting the existence of one problem in a certain part of the network can be useful enough for the user of the network. Additionally, it will be shown later that most unusual events do not share the same effects.

A. Kinds of Unusual Events

To discover the possible symptoms that a sensor network may suffer, it is first necessary to know what the existing

diseases are, that is, what we want the awareness mechanisms to detect. All unusual positions are triggered by one or more of the following principal causes: failure of a node, an external attack, or an internal attack. However, the diseases caused by attacks are far more numerous than the ones caused by node failure. There are many kind of attacks that can affect a sensor node, from the hardware layer to the application layer [8]. On the other hand, there are only two major events that the mechanisms should detect in case of node failure: a node that becomes unavailable from the network, and a sensor that malfunctions and provides inconsistent information. Therefore, in the remainder of this section, we focus on unusual events caused by external or internal attacks.

A malicious outsider with no prior knowledge of the network has two major objectives:

- To hinder the functionality of the network by affecting the physical environment or the communication channel.
- To tamper with (i.e., gain access to) one or more of the nodes in order to launch internal attacks.

Because the main task of a sensor network is to measure the surrounding phenomena, the adversary can try to fake the measurements taken by the sensors of a node. A simple attack is to directly manipulate the physical environment, such as submerging a node in water. However, a more stealthy attack is to substitute the sensors of a node with tampered sensors that provide erroneous data. This operation becomes easy if the sensors are simply plugged into the node or is moderately difficult if new sensors must be soldered.

The communication channel usually is protected by cryptographic primitives (e.g., the Advanced Encryption Standard [AES] cryptoalgorithm used in the IEEE 802.15.4 standard) and other mechanisms, such as timestamps and sequence numbers; thus, an adversary can try to jam only the signal. Jamming equals interfering with the radio frequencies used by the nodes or abusing the Media Access Control (MAC) protocol, disconnecting the nodes from the network as a result. These attacks to the communication channel and the physical environment or the sensors are somewhat effective, but an attacker could be more interested in accessing the security credentials contained inside the node. An attacker can access its hardware debug interface (e.g., JTAG) if it is not disabled or try to read the memory of the node in a non-trivial period of time [10]. Such an attack enables either the modification or cloning of the node.

After a malicious outsider has gained access to one or more of the sensor nodes, the attacker can manipulate the information flow that traverses the nodes. Therefore, it can perform internal attacks to the protocols of the network such as routing, aggregation, and time synchronization. The protocols of a sensor network usually are designed with a particular application in mind ([9]), so the scope and effects of these attacks depend on the specific protocol implementations used by the network. However, it is possible to classify the existing attacks that any reporting mechanism could partially detect into four attack templates:

- *Message creation* (related to malicious nodes creating fake packets regardless of the state of the other nodes in the network)
- *Packet alteration* (the contents of a relayed packet are changed in unacceptable ways)
- *Feature advertising* (a node broadcasts false control information)
- *Time-related attacks* (packets are delayed, selectively dropped, or do not reach their destination at all)

B. Position Awareness Mechanisms

After the diseases are known, it is possible to examine them to diagnose what their related symptoms are. That is, the analysis of the collateral effects of an unusual event will lead to the inference of the mechanisms that should be used to detect them. A summary of the different unusual events with their effects can be found in Table I. Note that in most cases, the detection mechanisms that infer the existence of unusual events are not complex, and such events can be detected just by storing and analyzing simple statistics generated by the network. As a result, these mechanisms can be lightweight enough for constrained environments such as WSNs.

A *jamming attack* is very difficult to circumvent, although it produces a clear symptom: an unusual decrease in the number of packets coming from the affected zone. Such symptoms can be detected by both the base station and the nodes on the routing path. Additionally, nodes belonging to or near the affected zone will detect an unusual increment of the number of collisions. Note that a single node that is not available due to *hardware failure* also will be detected by the base station and other nodes because of disappearing packets. However, in the case of hardware failure, there will be no unusual collisions in the neighborhood of the "dead" node.

TABLE I
RELATIONSHIP BETWEEN WSN ATTACKS AND THEIR SYMPTOMS

Abnormal event (disease)	Collateral effect (symptom)
Jamming	Wide data unavailability
Hw. failure ("unavailable" node)	Data unavailability
Node subversion	Node temporarily unavailable
Tampered, malfunctioning sensor	Deviations, inconsistencies
Message creation	Changes in packet density, inconsistent alerts
Packet alteration	Changes in packet (only for broadcast)
Feature advertising	Inconsistent feature with neighborhood
Time-related attacks	Long delays, traffic imbalance

A node will be temporarily unavailable from the network if an attacker is trying to *subvert* it. In this case, the number and ratio of messages from that node will drop to zero for a non-

trivial period of time. Therefore, a node that returns to the network after such a period of time has passed ([10]) should be considered suspicious by its neighbors and the base station.

A set of *false measurements* (either coming from a *tampered sensor* or a *malfunctioning* one) can be detected by the node itself, the neighborhood, and the base station. Certain values, such as the humidity of a room, do not fluctuate abruptly unless an extreme position (e.g., a flood) occurs, and the fluctuation continues over time. The neighborhood of a sensor node also should be able to sense the same physical readings if they are physically near. In addition, the base station might have a history of all the readings and could detect a significant deviation of the expected values based on the context and on the history of the network.

With regard to non-specific attacks against the core protocols of the network, we first consider *message creation*. Excluding alert and query messages, the nodes usually create and send packets to the base station only inside specific times frames (called *burst periods*). If the sensor nodes or the base station detect a change in the packet density of the network (i.e., more packets sent within a burst period), there is a chance that one of these attacks is taking place. Also, because an alert is referred to an event inside a physical area, nodes that route an alert and are close enough to the source node can check its validity. Additionally, if the base station does not issue a query to a certain region of the network, it is clear that no answer should come from that region.

Unfortunately, *packet alteration* attacks are very difficult to detect. The most obvious symptom is a change inside the information of a packet forwarded by a malicious node. However, in a sensor network with basic security services, the contents of a packet can be read only by its origin and its destination. Therefore, no one of the neighbors is able to read the contents of a relayed packet. However, there is a case in which this attack can be detected — broadcasted packets. They usually can be read by all members of the network, and a change can be detected easily. *Feature advertising* uses broadcast communication, too; thus all nodes in a neighborhood can check if the properties advertised by the source node have deviated from the reality of the network. For example, a node that is on the edge of the network cannot advertise that is near the base station.

Finally, a malicious adversary can execute *time-related* attacks by delaying, selectively forwarding, or dropping packets. With regard to *delayed packets*, it is atypical for a packet to be relayed later than the normal amount of time it may spend inside a normal sensor node under average stressful conditions. This deviation of the time for relaying a packet can be detected by nodes in the neighborhood by comparing the ratio of messages entering and exiting a certain node or the base station, and by comparing the time a packet takes to be routed from its source.

When packets are *selectively forwarded* (i.e., dropped) by a malicious node, it is obvious that these packets will not be received either by the next hop or by the base station. Nodes surrounding a malicious forwarder cannot verify if a specific packet was forwarded due to the protection of the communication channel. However, they may be able to check if there is an imbalance between the number of packets going to that node and the number of packets coming from that node. Finally, any node that *drops packets* relayed to it (a black hole) usually will not send a message, and such a piece of evidence easily can be detected almost immediately by any neighbor.

IV. A Primal Plan of an IDS Architecture for Wireless Sensor Networks

Position awareness mechanisms are essential to enable the monitoring of the elements of a WSN and the existence of the self-configuration property. Nevertheless, they also serve as the cornerstone for the development of IDSs for sensor networks. By knowing the position of its surroundings, a sensor node can decide whether a certain neighbor may be faulty or malicious and react accordingly. There are aspects related to IDSs that were discussed in previous works, such as the position of the detection agents, the nature of detection mechanisms, and so on. However, it is necessary to provide the primal plan of an IDS architecture for sensor networks. To improve the existing approaches, the architecture must fulfill all the following properties: full network coverage (cover the entire information flow of the network), simplicity (use mainly simple components, statistics, and mechanisms), usefulness (be able to detect all the standard positions where a neighbor may be behaving in a faulty or malicious way), extensibility (possibility to include new detection mechanisms), and inclusiveness (where all the existing research could be integrated).

For assuring full network coverage, a decentralized architecture must be used, because any part of the network can be a possible point of intrusion. As a result, the detection tasks must be performed by a software element (i.e., agent) located inside every node (*node agents*), and in every base station (*base station agents*). These two types of agents have different capabilities and use different sources of information. A sensor node is very constrained by nature, thus its node agent should employ only lightweight mechanisms. Also, the node agent can obtain information only from its direct neighborhood. On the other hand, the powerful base station receives information from all the nodes in the network, thus the base station agent can take advantage of this wealth of information to observe and analyze the behavior of its nodes.

The internal components of all agents are shown in Fig. 3. In our architecture, the *data acquisition* component obtains data from the sources of symptoms (e.g., packets and sensor information) and stores the processed information in the *statistics* component. These two components are used by the *detection* component that infers the existence of unusual events. This component can use both the position awareness mechanisms introduced earlier and other detection mechanisms that are part of existing or future research. All results are shared in the *alert database* component, where nodes are labeled as suspicious or malicious. Finally, the architecture includes a *collaboration* component that can be activated when the node must share an event with its other subsystems, its neighbors, or the base station.

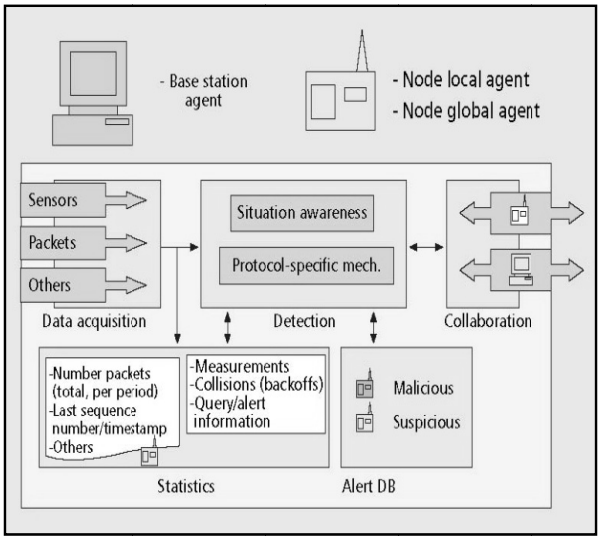

Fig. 3 Primal plan of an IDS architecture for WSN

The constraints inherent to the nodes impose the division of the tasks that are performed by a node agent. Consequently, this agent is composed by a *node local agent* that monitors only the information local to the node and a *node global agent* that can analyze the information flowing in its neighborhood. More specifically, node local agents are in charge of detecting unusual positions in both the specific protocols used in the network and in the sensor readings. Its detection mechanisms are executed whenever there is data available for analysis. On the other hand, to save energy, the detection mechanisms of node global agents are run at regular intervals (e.g., after the end of every burst period). These mechanisms can uncover the existence of jamming attacks, hardware failure, selective forwarding, and packet delay. Moreover, certain mechanisms (e.g., broadcast packet analysis) can be temporarily turned off due to the redundancy of sensor networks [6].

By including the detection mechanisms inside the same agent, it is possible to have a single source of information that can be shared by everyone. Also, due to the collaboration component, it is possible to improve the reliability of some detection mechanisms, such as the ones in charge of discovering selective forwarding attacks. Having this kind of architecture inside a sensor node does not pose a significant overhead: our prototype implementation in TinyOS 2.0, including the previously mentioned position awareness mechanism, fits in less than 4 kbytes of ROM and 500 bytes of RAM. As a final note, a concern may arise regarding the requirement of the global node agent to receive the packets from its neighborhood. However, the wireless nature of the communication channel forces it to do so, in order to check if it is the destination of the packet. While doing this checking, a node can update the statistics component (e.g., the number of packets sent by a node).

V. CONCLUSIONS

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 position awareness mechanisms. This work has shown how those mechanisms can be developed by considering a sensor network as a equilibrated organism where a deviation produced by a failure or by an attack will produce a detectable collateral event. This article used these mechanisms as a foundation for designing a primal plan of an IDS specifically designed for sensor networks. This system fulfills important goals such as total network coverage, simplicity, usefulness, extensibility, and inclusiveness. These goals are not met completely by the existing work in the field.

The mechanisms presented here are oriented to monitor networks that are static by nature. Actually, the most important applications of sensor networks, such as home automation, are built over these kinds of networks; thus the majority of the existing protocols and services are oriented to support only nodes that do not move from their initial deployment point. Nevertheless, applications with mobile nodes have a huge potential. Although the symptoms generated by an adversary or by a node failure in these mobile networks can be very different, it is possible to take advantage of the knowledge presented in this paper to define other position awareness mechanisms and IDSs that could work in mobile scenarios.

REFERENCES

[1] C. Hsin and M. Liu, "A Distributed Monitoring Mechanism for Wireless Sensor Networks," *Proc. 3rd ACM Wksp. Wireless Sec.*, Atlanta, GA, Sept. 2002.
[2] S. Rost and H. Balakrishnan, "Memento: A Health Monitoring System for Wireless Sensor Networks," *Proc. 3rd IEEE Conf. Sensor, Mesh, and Ad Hoc Commun. and Networks*, Reston, VA, Sept. 2006.
[3] I. Onat and A. Miri, "An Intrusion Detection System for Wireless Sensor Networks," *Proc. IEEE Int'l. Conf. Wireless and Mobile Computing, Networking and Commun.*, Montreal, Canada, Aug. 2005.
[4] P. Inverardi, L. Mostarda, and A. Navarra, "Distributed IDS for Enhancing Security in Mobile Wireless Sensor Networks," *Proc. 20th Int'l. Conf. Adv. Info. Networking and Apps.*, Vienna, Austrai, Apr. 2006.
[5] C. C. Su *et al.*, "The New Intrusion Prevention and Detection Approaches for Clustering-based Sensor Networks," *Proc. IEEE WCNC '05*, New Orleans, LA, Mar. 2005.
[6] R. Roman, J. Zhou, and J. Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks," *Proc. 3rd IEEE Consumer Commun. and Networking Conf.*, Las Vegas, NV, Jan. 2006.
[7] C. Basile *et al.*, "An Approach for Detecting and Distinguishing Errors versus Attacks in Sensor Networks," *Proc. 2006 Int'l. Conf. Dependable Sys. and Networks*, Philadelphia, PA, June 2006.
[8] J. P. Walters *et al.*, "Wireless Sensor Network Security: A Survey," *Security in Distributed, Grid, and Pervasive Computing*, Ed., Y. Xiao, CRC Press, 2006.
[9] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Commun.*, vol. 11, no. 6, Dec. 2004, pp. 6–28.
[10] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks," *Proc. 3rd Int'l. Conf. Sec. in Pervasive Comp.*, York, U.K., Apr. 2006.