R. Roman, J. Lopez, and S. Gritzalis, "Situation Awareness Mechanisms for Wireless Sensor Networks", *IEEE Communications Magazine*, vol. 46, pp. 102-107, 2008.
http://doi.org/10.1109/MCOM.2008.4481348
NICS Lab. Publications: https://www.nics.uma.es/publications

Situation Awareness Mechanisms for Wireless Sensor Networks

Rodrigo Roman[1], Javier Lopez[1], Stefanos Gritzalis[2]
[1] Computer Science Department, University of Malaga, Spain
{roman,jlm}@lcc.uma.es
[2] Department of Information and Communication Systems Engineering,
University of the Aegean, Greece
sgritz@aegean.gr

Abstract

A wireless sensor network is supposed to be able to operate for long periods of time with little or no external management. There is a requirement for this autonomy: the sensor nodes must be able to configure themselves in presence of adverse situations. Therefore, the nodes should make use of situation awareness mechanisms in order to determine the existence of abnormal events in their surroundings. This work approaches the problem by considering the possible abnormal events as diseases, thus making possible to diagnose them through their symptoms, i.e. their side effects. Considering these awareness mechanisms as a foundation for high-level monitoring services, this article also shows how these mechanisms are included into the blueprint of an intrusion detection system.

1 Introduction

The main purpose of a *Wireless Sensor Network* (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 one allows any user of the computer system access to the services provided by the sensor network. All data coming from the nodes, but also control commands directed to them, will traverse the base station.

Sensor nodes can be fully autonomous due to their battery-powered computational and communication capabilities. As a result of this autonomy, a sensor network is supposed to work without any human assistance during most of its lifetime. However, as a requirement for being 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. In fact, these mechanisms can serve as

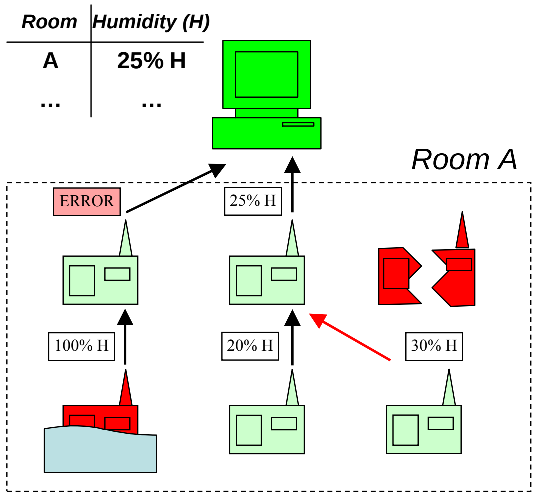


Figure 1: Importance of self-awareness mechanisms for sensor networks

a foundation for more complex schemes, such as *Intrusion Detection Systems* (IDS). IDS are particulary useful in scenarios where there is a chance that a node might be controlled by a malicious adversary.

This research work elaborates on the importance of mechanisms for detecting abnormal situations in sensor nodes, reviewing the main research activities in this area, and presenting a novel approach for the detection of events: considering a static WSN as a living body, an abnormal situation is seen as a disease, and associated with any disease there is a set of symptoms that can lead to its diagnosis. By analyzing both diseases and symptoms, it is possible to develop lightweight awareness mechanisms. We additionally highlight how it is possible to integrate those procedures into an IDS architecture.

2 Self-Configurability and Situation Awareness

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 preexisting 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 autonomy, sensor nodes are supposed to behave as self-configurable entities. They should be set up and deployed without any major effort by non-experts, and they should be able to adapt and heal themselves during the lifetime of the network.

However, in order to be fully autonomous 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 behaviour of the network. For example, the nodes that are affected when one of the routers of the network fails to work must be able to automatically notice it and react accordingly (cf. Fig. 1). The task of detecting such events relies upon the existence of situation awareness mechanisms. Without these mechanisms, a node can not fully understand the current situation of its environment, and will not be able to configure itself in order to respond to internal/external events. Note that these mechanisms have to be lightweight enough for allowing their execution in the constrained nodes.

There are some existing techniques that allow to control simple factors such as the actual situation of the sensor nodes. For example, the protocol [1] simply consists of sending periodical "heartbeat" messages to other nodes in order to check whether they are alive. It has been 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 abnormal situations caused by malicious nodes, either by analyzing the behaviour 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 *Intrusion Detection Systems* (IDS). IDS is an interesting — albeit underdeveloped — service, useful for scenarios where there is a chance of a node being subverted and controlled by an adversary. The major task of an IDS is to monitor computer networks and systems in order 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, though some authors have explored how to develop mechanisms for it.

Aside from the detection of abnormal events, there are other aspects in the development of IDS that need to be solved as well. The exact location of the detection agents and their tasks is an example. In hierarchical configurations, where more powerful devices named *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 needs some research. The redundancy of the network can be used as an advantage in this type of configuration, since (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 in order to detect malicious nodes and distinguish between possible errors and attacks launched against the network [7].

3 Development of Lightweight Awareness Mechanisms

As aforementioned, one of the key factors for the development of lightweight detection mechanisms is the acquaintance 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

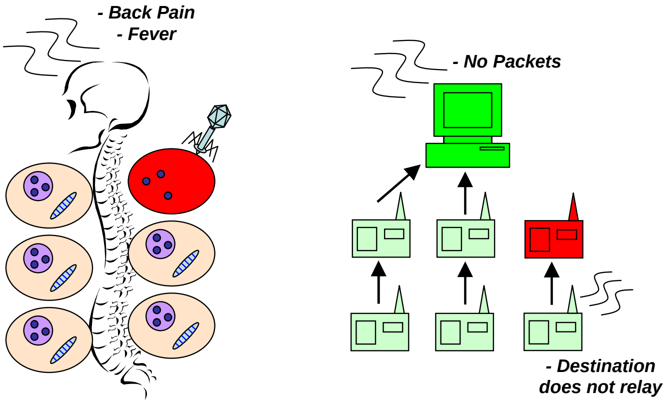


Figure 2: A sensor network as a living being

"brain", as seen in Fig. 2. Having in mind this simile, it is possible to think that the presence of certain symptoms (i.e. collateral effects) will be indicative of the existence of a disease (i.e. abnormal event). Therefore, the detection mechanisms will infer the existence of abnormal events based on the existence of their collateral effects.

One of the difficulties associated with the diagnosis of a disease consists on 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 all the 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 abnormal 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. Even more, it will be shown later that most abnormal events do not share the same effects.

3.1 Types of Abnormal Events

In order 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 abnormal situations are triggered by one or more of the following principal causes: failure of a node, a 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 abnormal events caused by external or internal attacks.

A malicious outsider with no prior knowledge of the network has two major objectives: (i) to hinder the functionality of the network by affecting the physical environment or the communication channel, and (ii) to tamper (i.e. gain access to) one or more of its nodes in order to launch internal attacks. Since 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 provides erroneous data. This operation becomes easy if the sensors are simply plugged into the node, or moderately difficult if the new sensors have to be soldered.

The communication channel is usually 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 only try to jam the signal. Jamming equals to 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 can 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 attack would allow him to either modify or clone the node.

Once a malicious outsider has gained access to one or more of the sensor nodes, it can manipulate the information flow that traverse them. 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 are usually designed with a particular application in mind (cf. [9]), so the scope and effects of these attacks depend on the specific protocol implementations used by the network. Still, it is possible to classify the existing attacks that any reporting mechanism could partially detect into four attack templates: (i) *message creation* (related to malicious nodes creating fake packets regardless of the state of the other nodes in the network), (ii) *packet alteration* (when the contents of a relayed packet are changed in unacceptable ways), (iii) *feature advertising* (when a node broadcasts false control information), and (iv) *time-related attacks* (related to whether packets are delayed, selectively dropped, or are not going to reach their destination at all).

3.2 Situation Awareness Mechanisms

Once the diseases are known, it is possible to examine them in order to diagnose what their related symptoms are. That is, the analysis of the collateral effects of an abnormal event will lead to the inference of the mechanisms that should be used in order to detect them. A summary of the different abnormal events alongside with their effects can be found in table 1. Note that, in most cases, the detection mechanisms that infer the existence of abnormal 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 WSN.

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, Inconsistences
Message creation	Changes in packet density, Inconsistent alerts
Packet alteration	Changes in packet (only for broadcasted)
Feature advertising	Inconsistent feature with neighborhood
Time-Related attacks	Long delays, Traffic imbalance

Table 1: Relationship between WSN attacks and their symptoms

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

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 (cf. [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 there is an extreme situation (e.g. a flood) going on, and that fluctuation should continue over time. The neighborhood of a sensor node should also be able to sense the same physical readings if they are physically near. At last, the base station may 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.

Regarding non-specific attacks against the core protocols of the network, first we 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 on the packet density of the network (i.e. more packets being sent within a "burst period"), there is a chance that one of these attacks is taking place. Also, since 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. Even more, if the base station does not issue any query to a certain region of the network, it is clear that no answer should come from that region.

Packet alteration attacks are, unfortunately, very difficult to detect. Its more 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 only be read by its origin and its destination. Therefore, no one of the neighbors is able to read the contents of a relayed packet. There is a case in which this attack can be detected, though: broadcasted packets. They can usually be read by all members of the network and any change can be easily detected. *Feature advertising* uses broadcast communication too, thus all nodes in a neighborhood can check if the properties advertised by the source node are too 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 some *time-related attacks* by delaying, selective forwarding, or dropping packets. Regarding *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 on 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 by the base station by comparing the time a packet needs 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 has been 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) will not send practically any message, and such piece of evidence can easily be detected almost immediately by any neighbor.

4 A Blueprint of an IDS architecture for Wireless Sensor Networks

Situation awareness mechanisms are essential in order to allow 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 IDS for sensor networks. By knowing the situation of its surroundings, a sensor node can be able to decide whether a certain neighbor may be faulty or malicious, and react accordingly. There have been aspects related to IDS that have been discussed in previous works, such as the situation of the detection agents, the nature of some detection mechanisms, and so on. Still, it is necessary to provide the blueprint of an IDS architecture for sensor networks. In order to improve the existing approaches, such architecture must fulfill all the following properties: full network coverage (cover all the information flow of the network), simplicity (use mainly simple components, statistics and mechanisms), usefulness (able to detect all the standard situations where a neighbor may be behaving faulty or maliciously), extensibility (possibility to include new detection mechanisms), and inclusiveness (where all the existing research could be integrated in).

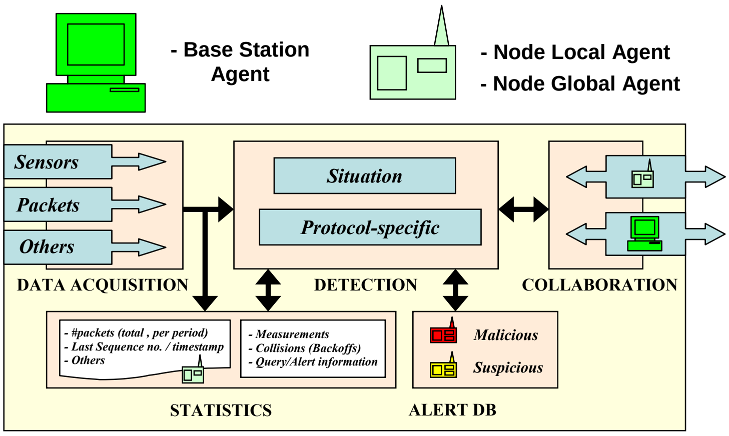


Figure 3: Blueprint of an IDS Architecture for WSN

For assuring full network coverage, it is necessary to use a decentralized architecture since any part of the network can be a possible point of intrusion. As a result, the detection tasks have to 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 only obtain information from its direct neighbourhood. 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 behaviour of its nodes.

The internal components of all agents is 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, which infers the existence of abnormal events. This component can use both the situation awareness mechanisms introduced in section 3 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 needs to share an event with other of its subsystems, its neighbors, or the base station.

The constraints inherent to the nodes imposes the division of the tasks that are performed by a node agent. Consequently, this agent is composed by a *node local agent*, which only monitors the information local to the node, and a *node global agent*, which can analyze the information flowing in its neighborhood. More specifically, *node local agents* are in charge of detecting abnormal situations 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, in order to save energy, the detection mechanisms of *node global agents* are run at regular intervals (e.g. after the end of every "burst period", cf. section 3.2). These mechanisms can uncover the existence of jamming attacks, hardware failure, selective forwarding, and packet delaying. Moreover, certain mechanisms (e.g. broadcast packet analysis) can be temporarily turned off, thanks 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, thanks 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 aforementioned situation awareness mechanism, fits in less than 4Kb of ROM and 500b of RAM. As a final note, a concern may arise on the subject of the node global agent having to receive the packets from its neighborhood. However, the wireless nature of the communication channel forces them to do so, in order to check if they are the destination of the packet. While doing this checking, a node can update the Statistics component (e.g. number of packets sent by a node).

5 Conclusions

Using its embedded sensors and the wireless channel, a sensor node can feel and interact with the world that surrounds it. Still, there is a difference between *feeling* the world and *understanding* the world. It is possible to shorten this gap using certain situation 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. Later, the article has used these mechanisms as a foundation for designing a blueprint of an intrusion detection system specifically designed for sensor networks. This system fulfills important goals such as total network coverage, simplicity, usefulness, extensibility, and inclusiveness. These goals are not completely fulfilled by the existing work in the area.

The mechanisms presented here are oriented to monitor networks that are static by nature. Actually, most important applications of sensor networks such as home automation are built over these kind of networks, thus the majority of the existing protocols and services are only oriented to support nodes that do not move from their initial deployment point. Nevertheless, application 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 from the knowledge presented in this paper to define other situation awareness mechanisms and intrusion detection systems that could work in mobile scenarios.

6 Acknowledgments

The authors wish to thanks Cristina Alcaraz, Gerardo Fernández and Miguel A. Ramírez for their useful comments during the development of this article. This work has been partially supported by the SMEPP project (EU-FP6-IST 0333563) and the CRISIS project (TIN2006-09242). The first author has been funded by the Ministry of Education and Science of Spain under the Programa Nacional de Formacion de Profesorado Universitario.

References

[1] C. Hsin, M. Liu. “A distributed monitoring mechanism for wireless sensor networks”. In Proceedings of the 3rd ACM workshop on Wireless Security (WiSe 2002), Atlanta, USA, Sep. 2002.

[2] S. Rost, H. Balakrishnan. “Memento: A Health Monitoring System for Wireless Sensor Networks”. In Proceedings of the 3rd IEEE Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2006), Reston, USA, Sep. 2006.

[3] I. Onat, A. Miri. “An Intrusion Detection System for Wireless Sensor Networks”. In Proceedings of the IEEE International Conference on Wireless and Monile Computing, Networking and Communicatons (WiMOB 2005), Montreal, Canada, Aug. 2005.

[4] P. Inverardi, L. Mostarda, A. Navarra. “em Distributed IDS for enhancing Security in Mobile Wireless Sensor Networks”. In Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Vienna, Austria, Apr. 2006.

[5] C. C. Su, K. M. Chang, Y. H. Kuo, M. F. Horng. “The New Intrusion Prevention and Detection Approaches for Clustering-based Sensor Networks”. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2005), New Orleans, USA, Mar. 2005.

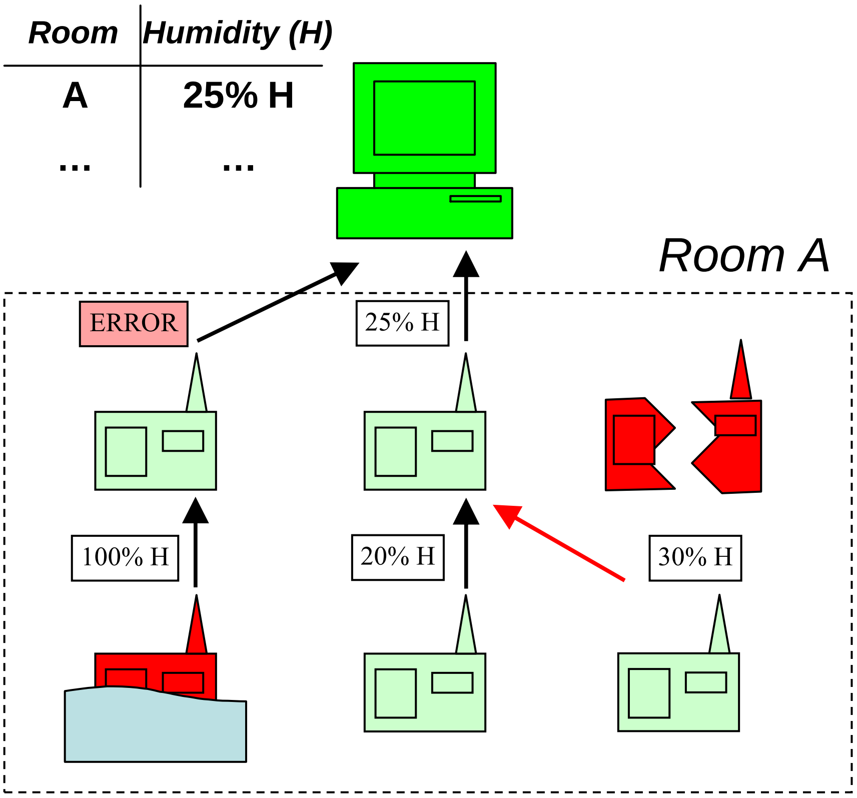
[6] R. Roman, J. Zhou, J. Lopez. “Applying Intrusion Detection Systems to Wireless Sensor Networks”. In Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC 2006), Las Vegas, USA, Jan. 2006.

[7] C. Basile, M. Gupta, Z. Kalbarczyk, R. K. Iyer. “An Approach for Detecting and Distinguishing Errors versus Attacks in Sensor Networks”. In Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN 2006), Philadelphia, USA, Jun. 2006.

[8] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary. “Wireless Sensor Network Security: A Survey”. Security in Distributed, Grid, and Pervasive Computing, Editor: Yang Xiao, Auerbach Publications, CRC Press, ISBN 0-849-37921-0, 2006.

[9] J. N. Al-Karaki, A. E. Kamal. “Routing Techniques in Wireless Sensor Networks: A Survey”. IEEE Wireless Communications, 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*. Proceedings of the 3rd International Conference on Security in Pervasive Computing (SPC 2006), York, UK, Apr. 2006.



Room	Humidity (H)
A	25% H
...	...

Figure (1): Importance of self-awareness mechanisms for sensor networks

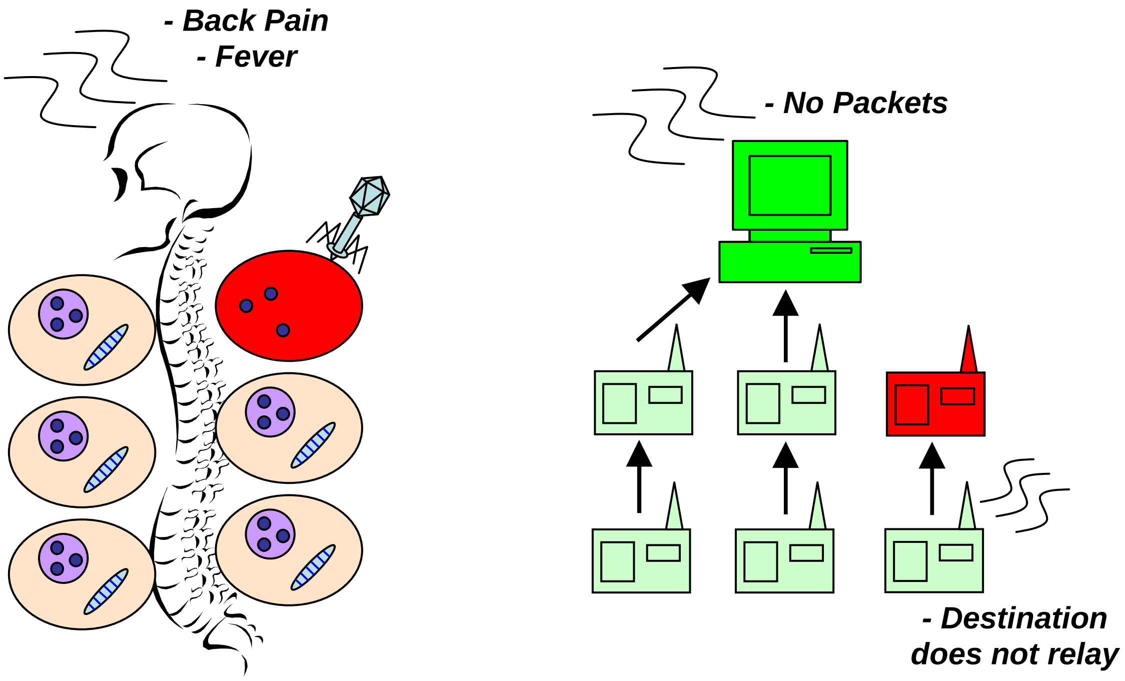


Figure (2): A sensor network as a living being

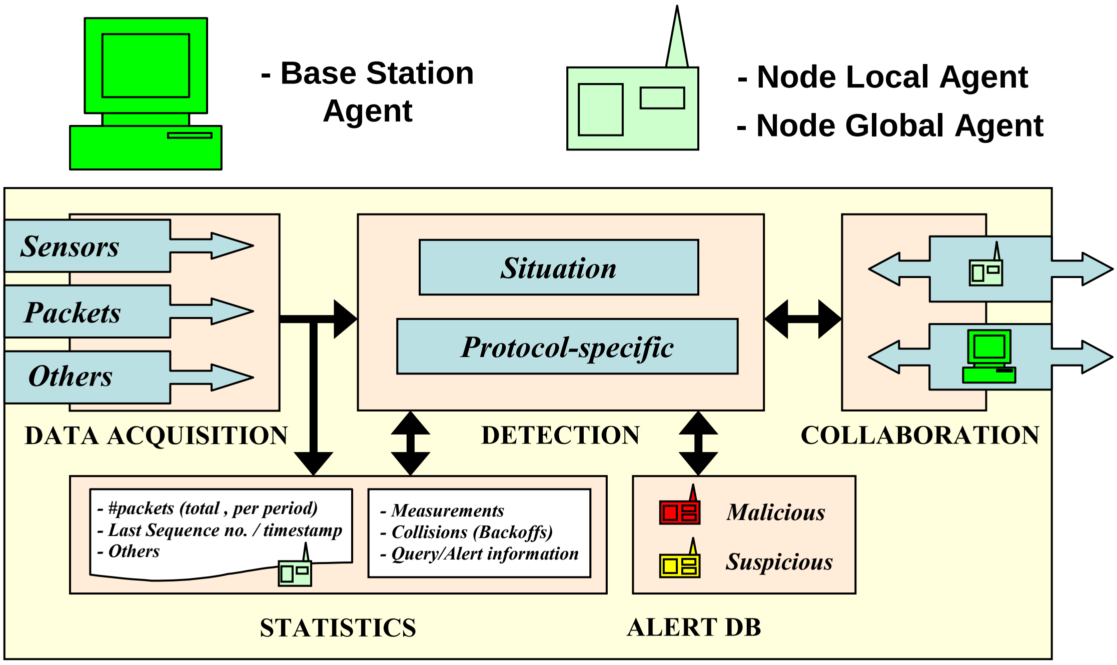


Figure (3): Blueprint of an IDS Architecture for WSN