SASHA: Toward a Self-Healing Hybrid Sensor Network Architecture

Tatiana Bokareva[1,2]
tbokareva@cse.unsw.edu.au
The University of NSW[1]
National ICT Australia Limited[2]

Nirupama Bulusu
nbulusu@cs.pdx.edu
Portland State University

Sanjay Jha[1]
sjha@cse.unsw.edu.au
The University of NSW[1]

Abstract

For widespread adoption of sensor technology, robustness in the event of abnormal behavior such as a network intrusion, or failures of components or nodes is critical. Current research on robust and resilient sensor networking is focused on specific tasks – secure broadcast, secure aggregation, secure localization or fault-tolerant feature extraction. While these primitives provide useful functionality, what has been lacking is a comprehensive, holistic *approach to sensor network robustness across various failure modalities.*

In this position paper, we propose a self-healing *hybrid sensor network architecture called SASHA, that is inspired by and co-opts several mechanisms from the Acquired Natural Immune System to attain its autonomy, robustness, diversity and adaptability to unknown pathogens, and compactness. SASHA encompasses automatic fault recognition and response over a wide range of possible faults. Moreover, it is an* adaptive *architecture that can learn and evolve its monitoring and inference capabilities over time to deal with unknown faults. We illustrate the workings of SASHA using the example of fault-tolerant sensor data collection and outline an agenda for future research.*

1. Introduction

Wireless sensor networks are a burgeoning focus of the research community, due to their potential to embed sensing and communication everywhere. One of the most valuable applications of wireless sensor networks is in the deployment of nodes in hostile or remote geographical locations, given the system's ability to operate unattended, without pre-existing infrastructure and with minimal or no maintenance[2].

Typically, sensor nodes are expected to be deployed randomly, organize themselves into a network, sense real world phenomena and forward observed measurements back to base stations. Due to their operational environment, sensor nodes are subject to frequent failures. For widespread adoption of sensor technology, it is critical that a system can heal itself in the event of abnormal behavior such as a network intrusion, or failures of components or nodes.

Previous work on robust and resilient sensor networking is focused on specific tasks, such as secure broadcast[15], secure aggregation[17], secure localization[14], or fault-tolerant feature extraction[13]. Such primitives constitute indispensable building blocks for sensor networks. But how should we use and combine these primitives? For example, sensor data can be corrupted by malicious sensors, as well as faults induced by physical world coupling. Sensor calibration techniques can compensate for faulty sensor readings, but cannot account for malicious aggregator nodes. Secure aggregation can protect against malicious aggregators, but cannot detect persistent faults in sensor data. Consequently, in each case, only one type of failure modality is addressed. In this case, what is required is a comprehensive approach to fault-tolerant data collection across various failure modalities. In general this motivates a whole-network approach to sensor network robustness that operates above the network layer.

This position paper proposes an immunology inspired solution to the design of a self-healing sensor network. We enumerate below our design goals for a self-healing sensor network architecture. We argue that these capabilities are inherent to the human immune system also advocated by Forrest[7].

1. *No Indispensability*. Failure of an individual component, node or communication link should have minimal impact on the entire sensor network operation. The human immune system is capable of replacing any of its basic cells. Similarly, no single node should be indispensable for the operation of the whole system.
2. *Autonomy*. In remotely deployed sensor networks, a large degree of operational independence is essential as the ratio of sensors to human is very high. There is no external entity responsible for the management of the Immune System, the self-healing process and elimination of pathogens happens independently.

3. *Several Layers of Protection and Detection*. Different mechanisms of a system provide different services. All of them are combined to provide an overall high level of detection and elimination of pathogens. In order to deal with a large number of possible faults in wireless sensor networks, support of multiple layers is needed.

4. *Compactness*. Our immune system can detect a large set of potentially harmful pathogens with a reasonably small number of detector cells. Our bodies cannot contain such a large number of lymphocytes and therefore it limits the minimum amount necessary. This is an essential quality for resource-constrained nodes in sensor networks.

5. *Diverse and Adaptive*. Our immune system can detect an incredibly large number of different types of pathogens. It has an amazing ability to detect previously unknown pathogens and remember them. This ability should be an important feature of wireless sensor networks, as the nature of possible threats cannot be known in advance.

The contributions of this paper are as follows. First, we motivate a holistic approach to sensor network robustness and propose a *self-healing sensor network* akin to the natural immune system. Second, to support the resource constraints of sensor devices, we propose a heterogeneous architecture called SASHA, where many different network entities coordinate with each other to provide efficient and effective sensor network fault-tolerance, encompassing automatic fault recognition, adaptive network monitoring and coordinated response (Section 3). SASHA is not intended to replace low-level security or fault-tolerance primitives, rather it provides a knowledge plane for the sensors to reason about and respond to various types of network failures, as illustrated in Figure 1. To ground our discussion, we illustrate how SASHA is envisioned to support fault-tolerant sensor data collection (Section 4). Finally, we present our agenda for future research (Section 5), and present our conclusions (Section 6).

2. Related Work

There are three avenues of focus directly related to this work (i) Sensor Fault-Tolerance (ii) Sensor Network Security and (iii) Local Area Network Security.

2.1. Sensor Fault-Tolerance

In the early 90s, Marzullo[12] was the first to address the problem of adapting to faulty sensor readings. The key idea is that if two sensors sample the same physical value, then their intervals must intersect. Marzullo's algorithms are centralized and not applicable to very large scale systems.

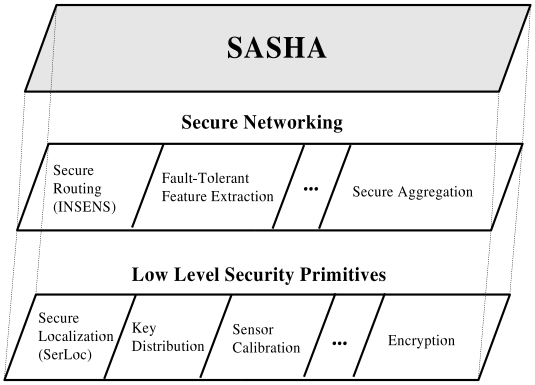

Figure 1. SASHA: A knowledge plane for a robust sensor network

In [6] authors develop a multi-modal sensing approach to fault-tolerance. If one type of sensor fails in the environment, the application can dynamically activate the other sensor. Their approach differs from ours in providing redundancy in sensor hardware in a single device, thereby increasing the cost and complexity of devices rather than exploiting the redundancy across densely deployed, simple devices.

Recently Krishnamachari and Iyengar [13] have proposed a solution to the recognition of faulty sensor readings, and introduced algorithms for self-organization which combine shortest-path routing, and the construction of a spanning tree as a clustering mechanism for nodes in a feature region. This work assumes that a simple threshold value is sufficient to determine the presence of an event, which may not be generalizeable to all events.

2.2. Sensor Network Security

The approach taken by the authors in [16] is to classify various types of data according to the sensitivity level and to identify possible communication security threats according to that classification. For each sensitivity level, distinct security mechanisms were proposed. One of the main principles stated in this work is that data items must be protected to a degree consistent with their value.

Wood and Stankovic[18] classify various types of denial of service attacks at different layers of the sensor network protocol stack and outline possible solutions related to communication security according to that classification.

Deng et al[3] describe the Intrusion-tolerant routing protocol for wireless sensor networks (INSENS). A major drawback is that communication between nodes in the network is only supported via a base station which introduces a large overhead in terms of packets and makes reprogramming of nodes over air a practically impossible task.

Two closely related research activities originated at UC Berkeley. Perrig et al [15] proposed a suite of security building blocks called SPINS: Secure Network Encryption (SNEP) and the micro version of the Time Efficient Streaming Loss-tolerant Authentication Protocol (μTESLA). In follow up work, Karlof et al[11] have introduced Tiny Security mechanisms (TinySec), which is the first implemented link layer security architecture for sensor networks. TinySec has been fully implemented on the TinyOS platform on Crossbow MICA hardware, and many of its features can be used in our self-healing sensor network implementation.

Ganeriwal and Srivastava [9] propose and simulate a reputation-based framework for high integrity sensor networks. Each sensor node assigns a reputation ranking to its neighbors, which characterizes them as cooperative or non-cooperative. The reputation can be assigned based on several factors, including data and routing consistency. Their work provides extensive algorithms for updating the reputation, taking into account information received from neighbors, but the question of what constitutes co-operative or non-cooperative behavior has been left largely unexplored. This is a major consideration in the design of our immune system based architecture, and the example in Section 4 provides an initial case study on the data consistency problem. Apart from a different view, our work also considers algorithms for checking data consistency.

2.3. Computer Immune System

Forrest et al first explored an immune systems approach to protect a local area network from network-based attacks. In one of the earlier works[1] Hofmeyer et al outlined a set of organizing principles and possible architectures for the implementation of the Artificial Immune System (AIS). Some of the design principles presented in this work are closely related to sensor networks.

In [8] Forrest and Hofmeyer outlined detailed descriptions of the AIS design applied to network security. The role of the AIS is to protect a Local-Area Network (LAN) from a network-based attack.

While Forrest's work is quite innovative, it is not directly applicable to sensor networks. Most of her work is developed for PC class devices and concentrates on the wired environment. As has been well documented, sensor nodes have significant computation, storage and energy constraints. Our self-healing sensor network must take into account the severe resource limitations of sensor devices.

We take a whole-network approach to a self-healing sensor network, wherein different network entities with varying resource and instrumentation capabilities, coordinate to automatically detect faults and provide a coordinated response to them. This requires a significantly distinct system architecture and creates different research challenges, as we discuss in the next section.

3. SASHA Architecture

In this section, we first describe (i) what constitutes a notion of *self* in sensor networks and (ii) a systems architecture that we envision. We then describe several specific problems that are the focus of our research and how they relate to our overall architecture.

3.1. Self in Sensor Networks

One of the main roles of the Natural Immune System is the recognition of self and the elimination of non-self proteins. In modeling an immune system equivalent for a sensor network, we must have a clear and stable definition of what constitutes the self and the non-self set. This is challenging for wireless sensor networks, because each application has its own unique characteristics and requirements. Nevertheless, we can identify several similarities that belong to an entire family of wireless sensor network applications.

Sensor nodes are usually deployed with a common goal in mind. Typically, the main role of the sensor nodes is to collect certain real world parameters and send them back to a base station. Given this, the correct sensor readings should reflect the behavior of an observed phenomenon. One of the most promising applications of sensor networks lies in their capacity to observe unknown environments and in such cases we may not have an exact knowledge of the phenomenon's behavior or have pre-collected sensor readings. How can we identify what constitutes correct sensor readings in such cases? In this paper, we approach the problem of identifying faulty sensor readings using pattern recognition techniques that leverage past observations of sensor nodes.

Sensors generate data streams. Sensor data can be characterized by continuous data streams. The pattern of data streams is neither common for all applications nor during different stages of an application. Some applications may require data to be sent on a periodic basis, whereas others may only require data to be transmitted in the presence of an event such as a burglar alarm. Therefore, another definition of self is an appropriate application behavior in terms of the periodicity of data delivery and the delivery of data from a set of authenticated nodes. Data integrity is closely related to sensor networks security. Sensor networks usually operate in an open environment. Nodes can easily be captured and the security information including cryptographic keys and functions can be easily recovered. Pair-wise cryptographic keys are one solution to avoid this problem, however the distribution and maintenance of such keys is a considerable challenge[4]. In summary the notion of self for a sensor network consists of (i) correct sensor readings, (ii) appropriate behavior of a running application event and (iii) authenticated set of nodes.

3.2. System Architecture

An example of our self-healing sensor network architecture is illustrated in Figure 2. It consists of several coordinating components, namely: a large number of sensing nodes, several monitoring nodes, base stations, Thymus, and Lymph (database) machines.

Sensing nodes: Sensing nodes are small, resource-constrained sensor nodes such as the Mica mote. They organize themselves into a network, sense and relay real-life measurements toward the closest monitoring nodes.

Some responsibilities of a sensor node as follows:

- Authenticate a set of neighbors.
- Authenticate packets received.
- Learn what constitutes the self-set in terms of sensor readings.
- Maintain connectivity to the monitoring node.
- Respond to the monitoring node's commands.

Monitoring Nodes: Monitoring nodes have enhanced sensing, processing and communication capabilities such as the Stargate. Each monitoring node covers a portion of the network topology. The sensor network will organize into a forest of trees, with each tree rooted at a monitor.

Some responsibilities of a monitor are:

- Authenticate the nodes in its tree as well as the neighbouring monitoring nodes.
- Monitor the behavior of its tree in terms of the noise level in the tree, the periodicity of data, the set of nodes, etc.
- Survey the sensor readings in a tree and ensure their correctness.
- Notify nodes of an appropriate action to be taken in case of an attack.
- Forward the data and attack notification to a base station.
- Query other monitoring nodes or base stations on the appropriate action to be taken in case of an attack or discovered anomalies.

Lymph: One of the major components of the natural immune system are B cells, a form of white blood cell. These cells are programmed to look for certain kinds of disease-causing pathogens, then destroy them and the cells infected by them. In a sensor network, to detect anomalies, the survey of a forest can be undertaken by means of mobile scripts running on all monitors, called B-script. A script is dynamically generated code and it acts as a filter for the behavior and statistical analysis of a forest. For example, scripts generated on a Lymph machine will have to reflect a non-self/malicious behavior of a forest. The Lymph machine serves as a database that will store signatures of past attacks, attacks scale of damage, the urgency required for a response and possible solutions to them. Mobile B-scripts will be dynamically generated and undergo positive selection on this machine. Most effective B-scripts will be issued to and run on the monitoring nodes.

Thymus: This machine is equivalent to humans Thymus and reserved for the representation of *self*. The role of the Thymus machine includes:

- Store representation of a self-set.
- Provide co-stimulation signals to the monitor to confirm the presence of faults.

Base Station: The role of the base-station is to provide a solution to the attack to monitoring nodes, and collect sensor data.

All these different entities are indispensable to SASHA. The system complexity and resource requirements increase progressively from sensing nodes, monitoring nodes, to base station, Lymph and Thymus machines.

3.3. SASHA Functions

SASHA has three specific aims:

1. *Automatic fault recognition* – Efficiently and automatically detect sensor faults.
2. *Adaptive network monitoring* – Efficiently evolve the monitoring and inference capabilities of the sensor network, so that it can adapt to a wide variety of unknown and unpredictable faults.
3. *Coordinated Response* – Network entities should coordinate and respond to various types of faults.

Automatic Fault Recognition.

To build a robust sensor network, we must foremost be able to recognize faulty sensor readings. To infer abnormal behavior, we can leverage the sensor data redundancy in densely deployed sensor networks and the statistical characteristics of the sensor data stream.

Automatic fault recognition consists of a lightweight, distributed learning algorithm to recognize faulty sensor readings from deviant characteristics in its self-set (Section 4).

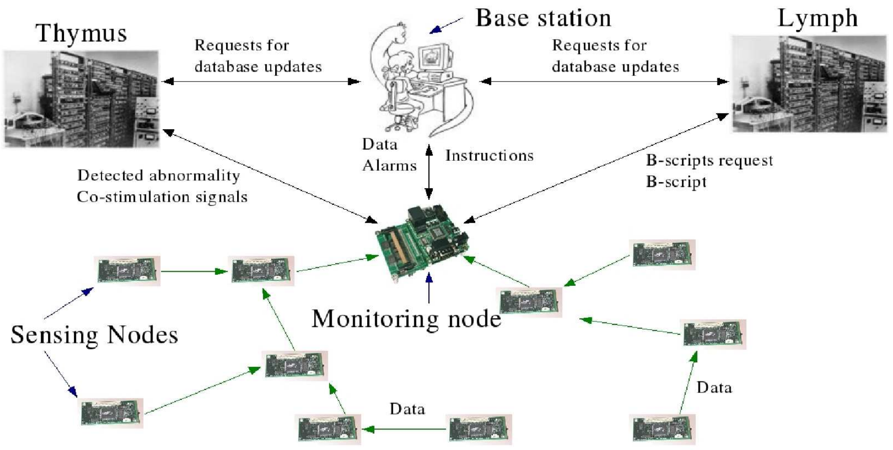

Figure 2. SASHA illustration for fault tolerant temperature data collection

Active network monitoring of the sensor network is accomplished by distributing mobile scripts onto the sensor network. This consists of the development of a system for generation, maturation and migration of mobile monitoring scripts. The generation of monitoring scripts should be autonomous and not involve any human intervention. This is challenging because of the following problems:

How do we distinguish normal and malicious behavior? We need to identify rules based on which a distinction between normal and malicious behavior of a forest can be made. Scripts running on a monitoring node will be generated based on these rules. We can use genetic algorithms to generate a continuously changeable set of scripts from existing descriptions of self and non-self sets in the Lymph and Thymus databases. Note that the Lymph and Thymus machines are envisioned to be PC class devices. A script can be encoded and represented as a string of 0,1 bits.

How do we generate and mature scripts? Evolution of the network monitoring scripts over time is a challenging problem. What is the right balance between exploration and exploitation? When promising possibilities are identified, they should be exploited at a *rate* and *intensity* related to their estimated *promise*, which is being continually updated. But at all times exploration for new possibilities should continue. The problem is how to allocate limited computation to different resources possibilities in a dynamic way that takes new information into account as it is obtained?

The immune system seems to maintain a near optimal balance between exploration and exploitation. At any time large numbers of B lymphocytes with different receptors are available for matching potential antigens; these different receptor types are formed via random combinations of genetic material in B cell precursors. In this way, the immune system uses randomness to attain the potential for responding to virtually any antigen it encounters. This potential is realized when an antigen activates a particular B cell and triggers the proliferation of that cell and the production of antibodies with increasing specificity for the antigen in question. Thus the immune system exploits the information it encounters in the form of antigens by allocating much of its resources toward targeting those antigens that are actually found to be present. But it always continues to explore additional possibilities that it might encounter by maintaining its huge repertoire of different B cells. The immune system combines randomness with highly directed behavior based on feedback. As in the immune system, in our architecture such an exploration strategy emerges from myriad interactions among simple, autonomous, and interacting components.

Coordinated Response *to malicious or faulty behavior is accomplished via coordination between monitoring nodes, or between monitoring nodes, Lymph and Thymus.* A monitoring node can raise an alarm by sending packets to other monitoring nodes in the network. The Thymus machine must provide a co-stimulation signal to a monitoring node. If a monitoring node does not receive a co-stimulation signal within a certain amount of time, it will deleted an associated script. On the other hand if it receives a co-stimulation signal from the Thymus machine, it can query the Lymph machine or a base station for further instructions.

4. Case Study: Fault-tolerant Data Collection

In this section, we describe the working of SASHA through a very simple example of building and maintaining a notion of *self* for the task of fault-tolerant collection of temperature measurements. This case study is focused on data consistency requirements, because the primary goal of all sensor network applications is to collect real world measurements. As was mentioned earlier much of the promise of sensor networks stems from their ability to monitor remote and unknown environments. In such applications we may have limited a priori information. In these kinds of scenarios, the usage of standard statistical approaches is a diffcult task because they mostly require offline processing of pre-collected sensor readings.

Imagine the scenario: A group of sensors such as Micas have been deployed to collect temperature samples. Suppose each group of 10 Mica nodes organized themselves into a tree rooted at a monitoring node. They take temperature measurements every minute and send these measurements to the monitoring node, as in Figure 2.

In order to identify faulty sensor readings, we need to model what constitutes the correct readings. We draw our inspiration from the field of Machine Learning. In particular, we use a *Self-Organizing Neural Network with Competitive Learning (SONN)* [5]. One of the main advantages of using SONN is that it does not require a priori knowledge of the phenomenon being monitored, just like the acquired immune system does not require a priori knowledge of pathogens. Therefore it can be applied to a larger set of sensor network applications than standard statistical approaches.

In order to classify its forest sensor readings as faulty or correct, the monitoring node will evoke the SONN that has been uploaded onto it from the Lymph machine. The competitive learning network divides a set of input parameters into data clusters and chooses the winning one. Figure 3 shows the collected temperature reading in a office, during a 24 hour period. Figure 4 shows the architecture of SONN used in this study. For n samples, SONN takes a $(n \times 10)$ matrix M, a (10×10) weight matrix W as inputs and produces a (1×10) vector F with its elements equals to the Eucledian distances between W and M.

$$F = \sqrt{\sum (W - M)^2} \tag{1}$$

The competitive layer returns a 1×10 vector C, with $0s$ for all neural inputs except for the closest element, which corresponds to a *winning* neuron. The output for the winning neuron is set to 1.

$$o_i = \begin{cases} 1 & \parallel w_i(t) - m_i(t) \parallel \leq \parallel w_o(t) - m_i(t) \parallel \forall o \\ 0 & \text{otherwise} \end{cases}$$

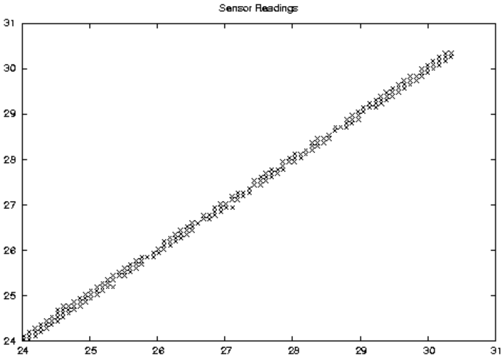

Figure 3. The temperature samples collected during a 24 hour period

The weight of the winning neuron is updated according to a simple *learning rule*

$$w_i(t+1) = w_i(t) + \lambda \times (m_i(t) - w_i(t)) \tag{2}$$

We set $\lambda = 0.01$ and the number of training epochs to 1000.

As a result of applying this *learning rule*, the weight of a winning neuron is updated to move closer to the corresponding input column of M. Eventually each cluster will output 1 if a similar vector is presented to SONN and 0 otherwise.

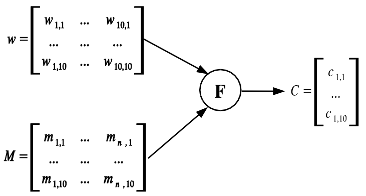
$w = \begin{bmatrix} w_{1,1} & \cdots & w_{10,1} \\ \cdots & \cdots & \cdots \\ w_{1,10} & \cdots & w_{10,10} \end{bmatrix}$
$M = \begin{bmatrix} m_{1,1} & \cdots & m_{n,1} \\ \cdots & \cdots & \cdots \\ m_{1,10} & \cdots & m_{n,10} \end{bmatrix}$
F
$C = \begin{bmatrix} c_{1,1} \\ \cdots \\ c_{1,10} \end{bmatrix}$

Figure 4. Self-Organizing Neural Network with Competitive Learning

This way SONN learns to categorize an input vector it sees. Figure 5 shows the associated path of weights during the learning period of 5 minutes.

Based on the SONN's output, the monitoring node can identify the most frequently winning cluster as the correct sensor readings. The resulting representation of *self* is a vector $v = [min_o, max_o]$. In this case, min_o and max_o are the values of a winning neuron, which are the minimum and the maximum temperature readings learned by this cluster.

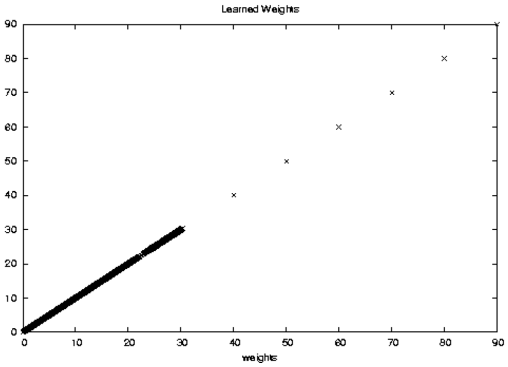

Figure 5. Final distribution of weights

During the training period, the vector v of the most frequently winning neuron is sent to the base station. Upon receiving this self representation, the base station will request an update of the database on the Thymus machine. The Thymus machine will update its database if the current self instance is not present. The database on the Lymph machine will also be updated with the corresponding representation of non-self.

Once the training period is over the Thymus machine will have the current representation of a *self set* related to the sensor readings. Correspondingly, the Lymph machine will have the current representation of a *non-self set*. The Lymph machine can now generate a monitoring script and send this script to the monitoring node. This script will survey outputs of the SONN in order to detect abnormal sensor readings. In our example an abnormal reading corresponds to a cluster that won the least number of competitions.

If an abnormality is detected, the corresponding vector v_k will be sent to the Thymus machine and the monitoring node will start a timer. The vector v_k is compared to other instances of self stored previously on the Thymus. If a match is not found then the Thymus machine will respond with the *co-stimulation* packet back to the monitoring node. This is necessary because the *self set* is dynamic and may change over time. It eliminates scripts that undergo maturation process on the ”old” non-self set.

Upon receiving the *co-stimulation* signal from the Thymus, the monitoring node will send this data to the base station marked as *false*. This is necessary in order to keep the representation of non-self up to date. It will also notify nodes about incorrect readings.

On the other hand, if the timer expires before the co-stimulation signal is received it is assumed that the monitoring script has ”binded” to self and it will be deleted. A new monitoring script will be requested from the Lymph machine.

Nodes in the tree will maintain a small internal state. This state consists of a counter that indicates how many times a node had consecutive faulty sensor readings. The count is augmented if a control packet received from the monitoring nodes matches the faulty sensor readings. Once the counter reaches a certain threshold, the node will request the retraining of SONN. If the control packet is not received, the counter is decreased at the next reading of the sensor value. By having a dynamical counter, we can distinguish a temporal noise in sensor readings from a permanent failure.

The ability to request retraining of SONN is an important feature. Most naturally occurring phenomena change over time and the representation of self in this case may not be accurate. However, the monitoring node will only initiate the retraining of the SONN (i) if more than 50% of its tree requests retraining and (ii) it has received confirmation from the base station. If the number of nodes requesting retraining is small, requesting nodes are assumed to be faulty and will be ordered to stop reading or sending their sensor values and act only as relay nodes or go to sleep.

5. Future Research

A complete realization of this architecture depends on several building blocks, including learning algorithms, coordination protocols, and genetic algorithms to support evolution. First, to support automatic fault recognition, we must evaluate SONN and its learning algorithm with regards to different application requirements. We plan to investigate the optimal frequency and duration of SONNs training period based on the complexity of the measured environment. Each application has its own specific requirements, such as the periodicity or the integrity of data. We have validated the SONN in a simulation environment and are currently working on its implementation on a testbed consisting of Mica’s and Stargates.

Second, appropriate abstractions for representing the self and non-self sets and the appropriate design choices for databases must be selected. In this paper, we introduced the representation of a self and non-self related to a physical sensors reading. However, SASHA is a module system and each representation of self is correlated to a particular fault. For example, representation of a faulty sensor reading will significantly differ from the representations of malicious sensor nodes and from the presence of DOS attacks. Both self and non-self sets are highly dynamic, thus, we must develop extensible and flexible data models for representation and storage of the self and non-self sets at the Thymus and Lymph respectively.

Third, we require efficient coordination protocols for construction and maintenance of a forest at the monitoring node, co-stimulation signals and interaction protocols between scripts and monitoring nodes, and an interaction protocol between the Thymus, Lymph and base stations. We plan to build on previous research in sensor network self-organization to construct efficient protocols for construction and maintenance of the forest[10].

Finally, we must employ genetic algorithms in service of a system for generation, maturation and migration of mobile monitoring scripts. Finally, we must conduct comprehensive performance evaluation of the system implementation to study both its efficiency and its effectiveness.

6. Conclusion

In this position paper, we proposed an immunology inspired self-healing sensor network architecture called SASHA. We illustrated how it could work using the case study of fault-tolerant sensor data collection. SASHA has several unique aspects that differentiate it from other ongoing work in robust and resilient sensor networking. First, it is a holistic approach to resilient network design. Instead of focusing on protocols to support specific security primitives, we develop a holistic system architecture that inspired by the human immune system, encompasses automatic fault recognition and response over a wide range of possible faults. Second, SASHA is an adaptive architecture that can learn and evolve its monitoring and inference capabilities over time. Several challenges must be met to achieve a complete realization of SASHA; nevertheless, we believe these capabilities constitute an important step toward robust and resilient sensor networking.

7. Acknowledgments

The motivation for this research emerged from discussions with Stephanie Forrest. This research is sponsored by the University of New South Wales, National Science Foundation (award CISE-RR-0423728), Tektronix, the PSU Foundation, and the Maseeh College of Engineering and Computer Science.

References

[1] S. H. Anil Somayaji and S. Forrest. Principles of a computer immune system. In *Proceedings of New Security Paradigms Workshop*, Cumbria, UK, November 1997.

[2] E. by D. Estrin and W. M. G. Bonito. Environmental cyber-infrastructure needs for distributed sensor network. *Scripps Institute of Oceanography*, Agust "12–14" 2003.

[3] J. Deng, R. Han, and S. Mishra. Insens: Intrusion-tolerant routing in wireless sensor networks. In *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003)*, Providence, RI, MAY 2003.

[4] D.Liu and P.Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the Conference of Computer and Communication Security(CCS'03)*, Washington D.C, USA, October 2003.

[5] R. D. E and Z. D. Feature discovery by competitive learning. In *Cognetive Science.*, volume 9, pages 75–112, 1985.

[6] A. S.-V. Farinaz Koushanfar, Miodrag Potkonjak. Fault tolerance in wireless ad-hoc sensor networks. In *Proceedings of IEEE Sensors 2002*, June 2002.

[7] S. Forrest and S. A.Hofmeyr. *Design Principles for the Immune System and Other Distributed Autonomous Systems*. edited by L.A. Segel and I. Cohen. Santa Fe Institute Studies in the Sciences of Complexity. New York: Oxford University Press, 2001.

[8] S. Forrest and S. Hofmeyr. Engineering an immune system. In *Submitted to Gaft*, February 2001.

[9] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 66–77, New York, NY, USA, 2004. ACM Press.

[10] W. Hu, N. Bulusu, and S. Jha. A communication paradigm for hybrid sensor/actuator networks. In *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC 2004)*, Barcelona, Spain, 5-8 September 2004.

[11] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004)*, November 2004.

[12] K.Marzullo. Tolerating failures of continuous-valued sensors. In *Proceedings of ACM Transactions on Computer Systems*, volume 8, no. 4, pages 284–304, November 1990.

[13] B. Krishnamachari and S. Iyengar. Efficient and fault-tolerant feature extraction in sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, Palo Alto, California, April 2003.

[14] L. Lazos and R. Poovendran. Serloc: secure range-independent localization for wireless sensor networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 21–30. ACM Press, 2004.

[15] A. Perrig, R. Szewczyck, V. Wen, D. Culler, and D. Tygar. Spins: Security protocols for sensor networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '01)*, pages 189–199, Rome, Italy, July 2001. ACM.

[16] S. S. M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B.Srivastava. On communication security in wireless ad-hoc sensor networks. In *Proceedings of the Eleventh IEE International Workshops on Enabling Technology:Infrastructure for Collaborative Enterprises(WETICE'02)*, 2002.

[17] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks. In *Proceedings of the ACM SenSys 2003*, Los Angeles, CA, November 2003.

[18] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(issue 10):48–56, Oct 2002.