A Dendritic Cell Inspired Security System in Wireless Sensor Networks

Jingjun Zhao
Department of Computer Science
North Dakota State University
Fargo, ND, USA
jingjun.zhao@ndsu.edu

Kendall E. Nygard
Department of Computer Science
North Dakota State University
Fargo, ND, USA
Kendall.Nygard@ndsu.edu

***Abstract* – We describe a scalable distributed methodology for increasing the rate of real packets received by the base station (BS) in a wireless sensor network (WSN) and to limit the inimical impacts of intruders in the network. The proposed security mechanism adopts a dual protection scheme to ensure that the BS obtains maximal real packets. First, we utilize a Dynamic Dendritic Cell Algorithm (DDCA) that effectively detects harmful intruders in a WSN and dynamically adjusts the monitoring period in response to the situation of the network. A sensor node running this algorithm can identify fake packets generated by the intruders based on pre-defined rules. Second, we apply a Markov Chain Monte Carlo (MCMC) method called the Metropolis-Hastings (MH) algorithm to infer the location of intruders in a wireless sensor network using partial information obtained from a subset of the sensor nodes. In turn these inferred locations are used by a fuzzy logic algorithm that we apply to assess the effect of the intruders on a monitored point. Based on the assessment, the BS sends commands that adjust the monitoring period that a sensor node uses to identify an intruder. The mechanism increases the flexibility and accuracy of the DC-inspired algorithm in accurately identifying harmful intruders, especially for harmful mobile intruders. The method can be applied to different sizes of WSNs and to both dynamic and static WSNs. We simulated the proposed algorithms using JADE (Java Agent Development Framework), and the results demonstrate good performance.**

Keywords-Wireless Sensor Networks; Dendritic Cells; Metropolis-Hastings algorithm; Fuzzy Logic; JADE

I. INTRODUCTION

A wireless sensor network (WSN) is composed of large numbers of small devices called sensor nodes that are typically deployed in an open and unprotected environment. They collect information and transmit data packets to a Base Station (BS). WSNs have been widely used in military and civil applications such as battlefield monitoring, environment and habit monitoring, and factory automation management. However, a WSN is susceptible to attacks and sensor failures such as packet dropping, packet change, energy-exhaustion, etc. Intrusion detection is an important research topic in a WSN, to help decrease power loss and to increase malicious event detection. Intrusion Detection methods utilized in wired networks are difficult to apply directly to WSNs because the sensor nodes are limited in battery power, storage, and computational ability. An Artificial Immune System (AIS) is a problem-solving methodology inspired by how biological immune systems in mammals are able to detect pathogens and destroy them before they cause harm to the body.

More specifically, Negative Selection Algorithm (NSA) has been used for solving different anomaly detection problems [14]. A NSA uses a learning phase to construct detectors which can identify and dispatch invaders, but are not harmful the organism itself. A fundamental issue in a NSA is that very difficult to maintain complete "non-self" detection in many real applications [13].

Following another type of AIS, the work in [4] advanced the Danger Theory (DT) approach to intrusion detection. Subsequent work by Nauman and Muddassar [6] established a security system based on the Dendritic Cells behavior. Work reported in [9] included detailed rules for the Dendritic Cell Algorithm (DCA) for analyzing abnormal signals. These DCAs are more flexible at detecting misbehaviors than NSAs, but do require monitoring period to identify an intruder. The period is fixed at the initial stage of a WSN and there is a tradeoff in deciding an appropriate monitoring period. A large period may lead to a low detection rate, and a little period may lead to a high error rate.

In our work, we employ a dual protection model that dynamically–adjusts to detect both static and mobile intruders while maintaining low energy consumption. Our DCA is primarily used to detect attacks: packet change, fake packet and energy-exhaustion. The ability to defend against these basic but widely existing types of attacks in a WNS makes our DCA a good fit in practice. To maximize the detection ability of our DCA, we also utilize the Metropolis-Hastings Algorithm (MHA) and fuzzy logic to dynamically adjust the monitoring period used in the DCA. The BS collects partial information about an intruders' location by requesting from a subset of the sensor nodes, then applies the MHA to estimate the location of the intruder. Because only a subset of the sensor nodes transmits location information, network longevity is enhanced. Upon acquiring the inferred information, the BS implements a fuzzy logic assessment algorithm that assesses the effect of these intruders on a monitored point. The assessment is sent back to the sensor nodes in this area, and these sensor nodes use it to adjust their monitoring period accordingly. This dynamic-adjusting mechanism ensures that the proposed DCA can accurately identify intruders and thus increases event detection reliability.

The rest of the paper is organized as follows: Section II discusses the related work; Section III describes the proposed DC-inspired algorithm, the usage of the Metropolis-Hastings algorithm, and the Fuzzy Logic algorithm; Section IV describes the designed multi-agent architecture for wireless sensor networks; Section V details our implementation and simulation results; finally, the conclusion and future work are given in Section VI.

 ISBN: 978-1-61208-107-6

II. RELATED WORK

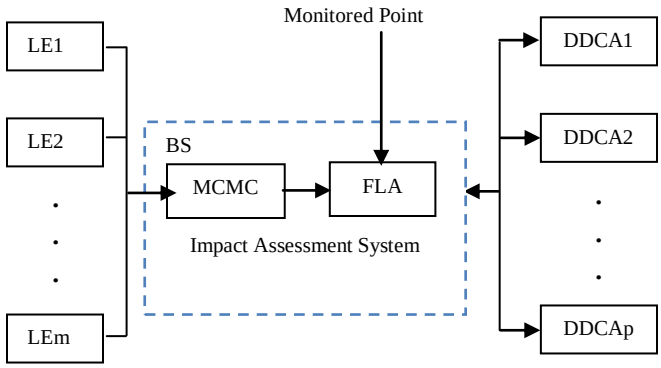
Greensmith, Aickellin and Cayzer [5] proposed a DC-based algorithm for the detection of anomalies. The authors categorize signals as Pathogen Associated Molecular Patterns (PAMPs), Safe Signals (S), Danger Signals (D), or Inflammatory Cytokines (IC). In [1], a description of the similarity between WSNs and AIS is provided, showing that a Dendritic Cell algorithm (DCA) can detect Cache Poisoning attack in a WSN. In this work, each node has an interest cache with fixed size that is used to record received history interest packets. Newly received packets replace older ones in the interest cache when the cache is full. These history packets are used to help a node decide whether or not to drop a received packet. The Ubiquitous Dendritic Cell algorithm (UDCA) combines the interest cache and the data cache in a node to analyze and identify danger signals. This algorithm can potentially detect an interest cache poisoning attack at an early stage. In [4], the authors employ Danger Theory (DT) in an intrusion detection system. Nauman and Muddassar [6] built upon an existing AIS-based security system described in [7] by simulating the behavior of a DC. In this approach, danger signals lead to the maturing of the DCs. The matured DCs activate detectors, which are then used to detect "non-self" network nodes. The work in [8] introduced a Dendritic Cell Algorithm. In [9], rules are formulated for identifying intrusion in a WSN.

The fixed cache size and monitoring period assumed in these algorithms will restrict their effectiveness in some applications. For example, these algorithms will have less efficiency on detecting mobile intruders than detecting static intruders in a WSN because mobile intruders can move out from the surveillance range of a sensor node before being detected. In this paper, we propose a Dendritic Cell algorithm that dynamically adjusts the monitoring period in a WNS. Our algorithm has great flexibility when compared with these existed algorithms.

III. THE DENDRITIC CELL APPROACH

A. Dendritic Cells

In a biological immune system, Dendritic Cells (DCs) are considered to be the most important Antigen Presenting Cells (APCs). Their basic role is to mark the surface of harmful antigens so that they can be recognized and destroyed by other cells in the immune system. The DCs are derived from bone marrow cells and begin their existence in an immature state. As an immature DC collects surrounding antigens, it is transformed to a semi-mature state. Eventually a semi-mature DC can receive sufficient danger signals from antigens to transform to a fully mature state at which time it is capable placing the danger marks on the surface of the harmful antigens. A Dendritic Cell Algorithm (DCA) is a problem-solving method that is based on the behavior of dendritic cells. We have designed a new DCA used to detect harmful intruders in a WSN. An innovation in our method is that the, monitoring period is dynamically adjusted by the base station in accordance with the current network status. Thus, we refer to the method as a Dynamic Dendritic Cell Algorithm (DDCA). To achieve the dynamic behavior, we employ a Markov Chain Monte Carlo (MCMC) technique to infer the distribution of the intruders in the network, then apply a Fuzzy Logic Algorithm (FLA) to assess the impact of these intrudes on a monitored point. Figure 1 shows the integration of the DC-inspired detection system and impact assessment system. We describe these three algorithms in more detail in the following sections.



m: the number of sensor nodes that are queried by the BS
p: the number of sensor nodes around the monitored point
LEi: the location of intruder i
DDCAi: the sensor node i running DDCA around the monitored point

Figure 1. The block diagram showing the integration of Dynamic Dendritic Cell Algorithm (DDCA), MCMC and Fuzzy Logic based Algorithm (FLA)

B. The Dynamic Dendritic Cell Algorithm (DDCA)

In this paper, we primarily recognize the following types of attacks in a WSN:

- the transmitting of changed messages;
- the reporting of messages at a frequency different than normal or expected ; and
- the reporting of fake messages.

When a sensor node sends out a message, we assume that all the neighbor nodes of the sender can receive and interpret the message. Each sensor node has the ability to discern if a neighbor node is transmitting a suspicious message (such as a changed or false message). A sensor node flags a harmful intruder if the number of suspicious message exceeds a pre-specified threshold value.

Sensor nodes are typically battery powered and have limited energy that must be used efficiently. The energy consumption rate for communication greatly exceeds that for sensing. Thereby, the useful lifetime of a WSN is largely governed by the management of the transmitting and receiving of messages. Ignoring messages rather than communicating with harmful intruders is energy conserving.

In the first phase of identifying a harmful intruder, suspicious messages are flagged by placing the senders onto a semi-harmful intruder (SHI) list. Additional detailed monitoring of subsequent message traffic will trigger the placing of the node onto a harmful intruder (HI) list. If an innocent period has elapsed, a node is deleted from the SHI list. Figure 2 illustrates the structure of this algorithm.

 ISBN: 978-1-61208-107-6

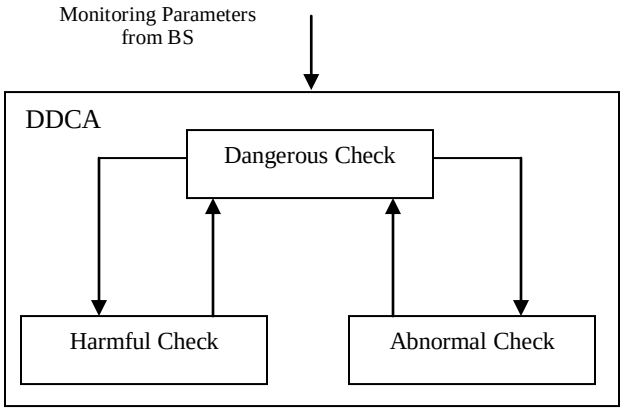


Figure 2. The Structure of Dynamic Dendritic Cell Algorithm (DDCA)

The following code in Figure 3 describes the DDCA in detail.

```
m_n: new message
HI: Harmful Intruder
SHI: Semi-Harmful Intruder
T_1: the maximum time slice calculating abnormal frequency
     of reporting message
T_2: the maximum time slice supervising a suspicious intruder
t_0: threshold for reporting message frequency during T_1
t_1: threshold for reporting message frequency during T_2
t_2: threshold for reporting false message rate during T_2
t_3: threshold for reporting changed message rate during T_2
 (Note: to tolerate casual high frequency of reporting
message, let T_2 > T_1)

//To check if the received message is dangerous
BOOL CheckDangerMessage (m_n)
    IF (m_n came from HI)
        return TRUE
    ELSE IF (m_n came from SHI)
        IF (CheckHarmfulIntruder(m_n) == TRUE)
            return TRUE
    ELSE IF (CheckAbnormalMessage(m_n) == TRUE)
        update SHI list
    ELSE
        return FALSE

//To check if the received message is abnormal
BOOL CheckAbnormalMessage (m_n)
    IF (m_n is a changed original message)
        return TRUE
    ELSE IF (m_n is a fake message)
        return TRUE
    ELSE IF (the frequency of the sender reporting
            message > t_0)
        return TRUE
    ELSE
        return FALSE

//to decide if the message sender is an harmful intruder by
checking the history records
BOOL CheckHarmfulIntruder (m_n)
    update the history message record of the sender
    estimate the frequency and percentage of the sender
    IF ((the new frequency > t_1) or (the new percentage
    of false message > t_2) or (the new percentage of
    changed message > t_3))
        clear the history record of the sender
        move the sender from SHI list to HT list
        return TRUE
    ELSE IF (the supervised time of the sender > T_2 )
        remove the sender from SHI list
    ELSE
        return FALSE
```

Figure 3. The Dynamic Dendritic Cell Algorithm (DDCA)

This algorithm consists of three sub-functions: checking for dangerous messages, abnormal messages and harmful intruders. Figure 2 presents the relationships among these sub-functions. When a sensor node receives a new message, the algorithm first checks the HI list. If the sender of this message is an identified harmful intruder, the message is considered to be a dangerous message. If the sender is an identified suspicious intruder that exists in the SHI list, the algorithm runs the function for identifying harmful intruder to decide if this sender is dangerous enough to be considered harmful. Finally, the algorithm runs the function of checking for abnormal message and possibly puts the sender on the SHI list. This algorithm will only consume limited energy of a sensor node because of the simple calculation in each sub-function. The dynamic adjustment of parameters increases the flexibility and accuracy of the DC-inspired algorithm in detecting dangerous messages.

C. The MCMC Method

We employ a Markov-chain Monte-Carlo (MCMC) method to infer the location of an intruder. The MCMC approach is based upon the Metropolis-Hastings algorithm. Figure 4 describes the general process of this algorithm. By recognizing the location of intruders, the BS can do a more effective assessment of received packets. If the BS were to send query messages to all sensor nodes in the WSN to collect location information, excessive energy would be required. The MCMC technique infers the intruders' location from limited local information. We assume that the fixed intruders are uniformly distributed over the network in the beginning. To the proposal distribution, we let the random location of an intruder moving to is chosen uniformly. That means the movement of an intruder is conditionally independent, which is a necessary condition for running the Metropolis-Hastings algorithm. In the training phase, a sensor node reports a message to the BS when an intruder is identified. The BS obtains the intruders' initial location from these messages.

 ISBN: 978-1-61208-107-6

Initial x_0

For i = 0 to n-1

Sample $u \sim u[0,1]$

Sample $y \sim q(y|x_i)$

$\alpha(x_i, y) = \min\{1, \frac{\pi(y)q(x_i|y)}{\pi(x_i)q(y|x_i)}\}$

If $(u < \alpha(x_i, y))$

$x_{i+1} = y$

Else

$x_{i+1} = x_i$

Figure 4. The Metropolis-Hastings algorithm

Figure 4 shows that a proposal movement is accepted if the calculated probability α bigger than a random number μ uniformly drawn from between 0 and 1. To minimize the complexity of the calculations, we use the method of [12] shown below for the acceptance criterion α.

$$\alpha(LE^t, LE^{t+1}) = \min(1, \frac{P(LE^{t+1}, R_1 \dots R_P)}{P(LE^t, R_1 \dots R_P)}) \quad (1)$$

LE^t: the vector of the intruders' location at time t
R_i: the ith sensor measurement
p: the number of nodes that the BS requests

Simplifying the following expression

$$\frac{P(LE^{t+1}, R_1 \dots R_P)}{P(LE^t, R_1 \dots R_P)} \quad (2)$$

to

$$\prod_{k=1}^{K} \frac{P(R_{K_k}|LE^{t+1})}{P(R_{K_k}|LE^t)} \quad (3)$$

K: the number of affected sensor nodes
K_k: the index of the kth affected sensor

In equation (3), there is only one variable in vector LE that has changed value from step t to step t+1. This change only affects those sensor nodes that can sense the change. This equation makes the calculation of the acceptance criterion easy and fast. The detail process of simplification can be found in [12].

D. The Fuzzy Logic Algorithm

Once the location of the intruder is known, a fuzzy logic algorithm is used to calculate the impact of these intruders on the accuracy of the packets received by the BS from designated areas in the network.

1) Inputs and outputs

a) The fuzzy algorithm has three inputs

Each input is classified into three categories: small, middle and large. The threshold values can be adjusted according to the requirements of different applications.

- Distance – the distance between an intruder and the monitored point
  Small: distance ≤ 5m
  Middle: 5m< distance < 15m
  Large: distance ≥ 15m
- Danger degree – classifying intruders based on their danger attacking type, i.e.,
  Small: reporting false message;
  Middle: transmitting changed original message;
  Large: reporting message in a frequency higher than expected one;
- Relative Position – using the angle of an intruder, the monitored point and the BS to represent the relative position of an intruder and the BS
  Small: $\theta \leq 20^0$
  Middle: $20^0 < \theta < 45^0$
  Large: $\theta \geq 45^0$

b) The fuzzy algorithm has five outputs:

(0, 0.25, 0.5, 0.75, 1)

These outputs represent different impacts of the intruders on the monitored point.

2) The membership functions

Figures 5 and 6 present the member functions of the fuzzy logic algorithm. The outputs of membership functions are defined as: small, middle and large, depending on the distance and the relative position individually.

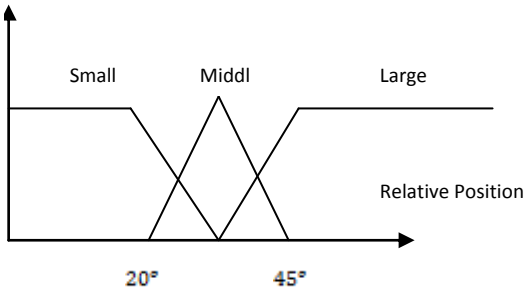


Figure 5. The input is the relative position.

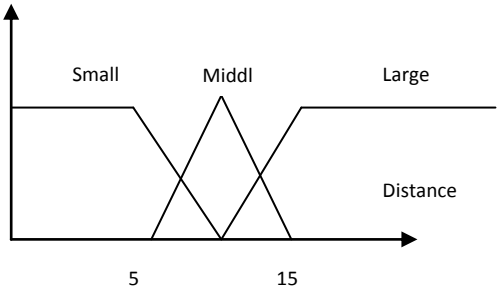


Figure 6. The input is the distance.

3) Fuzzy Rules

The fuzzy rules are listed in Table I. Shorter distance, smaller relative position, and consuming more energy will result in a classification of danger.

TABLE I. FUZZY RULES

Inputs			Output
Distance	*Danger Degree*	*Relative Position*	
S	M/S	M/S	0.25
S	M/L	S	0.5
S	L	L	0.75
S	L	M/S	1
M	M/S	M/L	0.25
M	L	M	0.5
M	L	S	0.75
L	-	M/L	0
L	-	S	0.25

4) Usage of the Fuzzy Logic Algorithm

The fuzzy logic algorithm is used to assess the impact of the intruders on a monitored point in a sensor network. To calculate the impact, we define the following equation:

$$\overline{IA} = \frac{\sum_{i=1}^{n} IA_i}{N}$$

$\overline{IA}$: the assessed impact of intruders

IA_i: the impact of the ith intruder (i.e., the output of the fuzzy logic algorithm)

n: the number of intruders

N: the number of sensor nodes around the monitored point

The BS in a WSN will adjust the monitoring parameters T_1 and T_2 used in the DDCA algorithm, according to the new assessment. Because the value N is fixed, more intruders will increase the assessed impact. Our rule is that the bigger the assessment is, the smaller the monitoring parameters become.

IV. MULTI-AGENT ARCHITECTURE

Using Multi-agent Systems (MASs) to model WSNs is considered to be a powerful and flexible approach [11]. The work reported in [10] concerns implementing a MAS using JADE (Java Agent DEvelopment Framework), a software framework fully implemented in JAVA. Similarly, we use JADE for our work. We have identified a set of common functions in WSNs and map each function with an agent, as shown in Table II. The architecture is composed of five types of agents: the Central Control Agent (CCA), the Sensing and Transmitting Message Agent (STMA), the Environment Agent (EA), the Message Analysis Agent (MAA), and the Immune System Agent (ISA). Figure 7 illustrates the designed Multi-Agent Architecture.

TABLE II. MAPPING BETWEEN AGENTS AND FUNCTIONS IN A WSN

Agents	Functions in a Wireless Sensor Networks (WSN)
The Central Control Agent (CCA)	Control center in a WSN
The Sensing and Transmitting Message Agent (STMA)	A sensor node
The Environment Agent (EA)	Environment of sensor nodes deployed
The Message Analysis Agent (MAA)	Sense of abnormal messages
The Immune System Agent (ISA)	Intruder detection

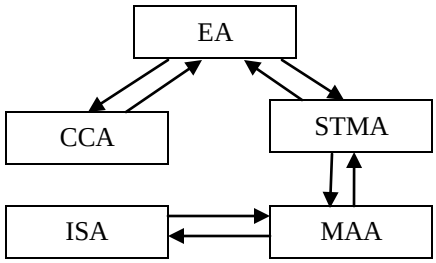


Figure 7. The designed Multi-Agent Architecture

The CCA sends control commands to the STMA. The STMA collects information and sends it to the CCA. The EA provides an operating environment for the other agents. The MAA is used to judge if a node is a potentially harmful intruder. The ISA is responsible for recognizing harmful intruders. The ISA is activated only when a message is recognized as a suspicious message by the MAA. The detailed functions of the agents are listed in Table III We run the proposed algorithm in the ISA to discriminate between normal messages and abnormal messages.

TABLE III. AGENTS' RESPONSIBILITIES

Agent type	Responsibilities
The Central Control Agent (CCA)	Sends commands to the EA Receives messages from the EA
The Sensing and Transmitting Message Agent (STMA)	Sends messages to the EA. Presents incoming messages from the EA Allows a sensor to accept incoming messages Allows a sensor to reject incoming messages
The Environment Agent (EA)	Presents incoming messages from the CCA and the STMA Sends the received messages to the CCA and the STMA
The Message Analysis Agent (MAA)	Sends messages to the STMA Sends messages to the ISA Recognizes abnormal message
The Immune System Agent (ISA)	Sends messages to the MAA Recognizes harmful intruders

V. SIMULATION RESULTS

We evaluate the performance of the security scheme using simulation. We implement the algorithms in JADE, which is an agent platform that is compliant with the Foundation for Intelligence Physical Agents (FIPA) standards for multi-agents. Our multi-agent architecture is similar to that reported in [10], which supports differing types of agents and inter-agent communication. Each sensor node agent hosts the DC-inspired algorithm, and the BS agent hosts the MH algorithm and the fuzzy logic algorithm.

The test area is a 100 by 100 square region, and 100 sensor nodes are randomly deployed in this area. Our experiments use a fixed number of intruders randomly deployed in the test area.

We conducted three groups of experiments to evaluate the efficiency of the dual protection scheme in a sensor network. The first group of experiments is executed to estimate the efficiency of the proposed DC-inspired algorithm on

 ISBN: 978-1-61208-107-6

maintaining the network lifetime. The second group of experiments is performed to measure the effect of cache size on the intruder detection rate. The last group of experiments is conducted to assess the performance of the dual protection scheme on detecting mobile intruders in a WSN. The intruders in the sensor network are static in the first two groups of experiments and mobile in the last one. The total number of sensor nodes is constant in all experiments.

A. *Maintaining the network lifetime*

In this group of experiments, there are 100 sensor nodes in the test area. When a sensor node runs out of battery life, normally or abnormally, it becomes a failed sensor node. We assume that a prescribed number of failed nodes will cause network failure and treat this as the only factor affecting the network lifetime. We set the number of failed nodes that cause the network failure to 70. To estimate the efficiency of the DC-inspired algorithm, we sample the values of failed sensor nodes at different time slots, and compare the results with an experiment that simulates our DC-inspired algorithm and a parallel experiment in which no security mechanism is employed. We randomly deploy 10, 20 and 30 intruders in the test area and perform experiments on each case. The results show that the DC-inspired algorithm greatly increase the network lifetime when compared with no security mechanism employed. Figure 8 shows that the network lifetime is doubled. Figures 9 and 10 show a longer network lifetime despite more intruders added. It is related to the final distribution of intruders.

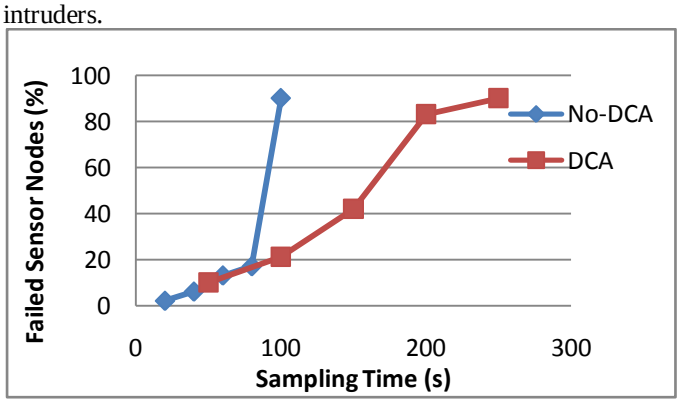


Figure 8. Sampling Time vs Failed rate of 10 intruders

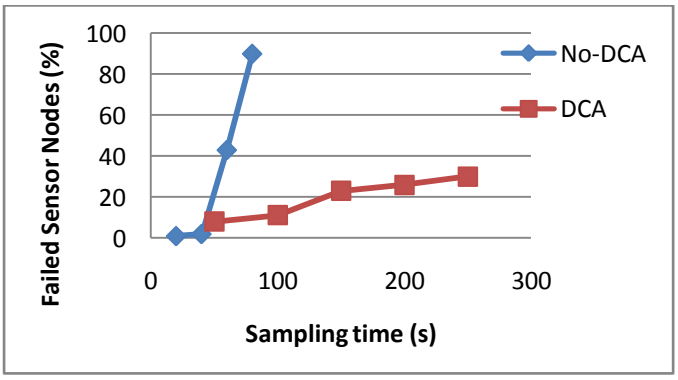


Figure 9. Sampling Time vs Failed rate of 20 intruders

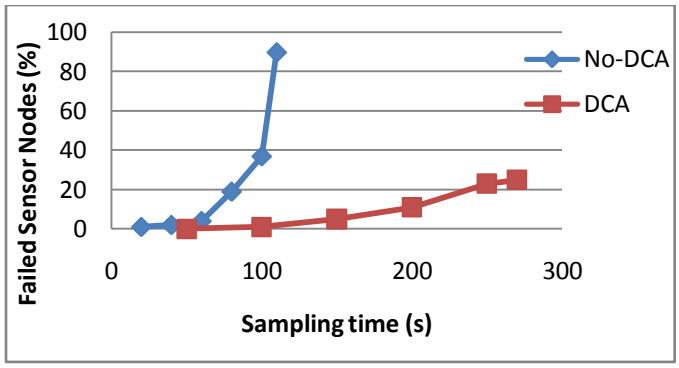


Figure 10. Sampling Time vs Failed rate of 30 intruders

B. *Cache Size Effect on the Detection Rate*

In the second group of experiments, we keep the same size of the test area and the network size as in the first group of experiments. Our aim is to measure the effect of the cache size on the detection rate of the DC-inspired algorithm. To detect a harmful intruder, a sensor node uses a cache to store the abnormal packets received from an intruder, and to monitor the intruder for a period of time to assess whether or not this is a harmful intruder. We set the cache size to 50, 100, 200, 500 and 1000 units and perform experiments on each cache size.

Figure 11 shows the experimental results. As expected, a larger size cache uniformly has a higher detection rate than a small cache. There is a flattening of the curves after a cache size of 100, indicating only marginal benefits of caches of size 200 or larger. When there are 10 intruders, the algorithm detect almost all of the harmful intruders with a cache size of 1000. The results also show that the procedure achieves a uniformly better detection rate for smaller numbers of intruders than for larger numbers, for any of the cache sizes. This is because the packets that a sensor node receives from harmful intruders are saved in the shared cache, and, when the cache is full, old packets will be removed from the cache, even though these packets may still be useful for detecting an intruder. Hence, the algorithm has a lower detection rate when more intruders exist in the network. These experiment results indicate that cache size is an important factor for detecting harmful intruders. However, large cache sizes may be unrealistic in resource-limited low-end sensors.

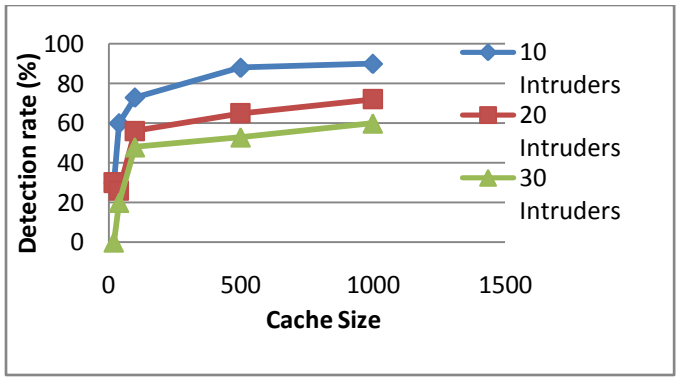


Figure 11. Cache Size vs Detection Rate

 ISBN: 978-1-61208-107-6

C. Assessment of the Dual Protection Scheme

The third group of experiments is designed to assess the performance of the proposed dual protection scheme. In these experiments, a fixed number of mobile intruders are randomly deployed in the sensor network. We measure the accuracy of the received packets by the BS to evaluate the efficiency of the scheme. At the beginning of the experiments, a monitored point is randomly chosen in the test area. In a real application, the BS could specify the monitored point according to the actual network topology. In our experiments the BS knows the number of intruders and their initial location, which is necessary condition in the MH algorithm. In a real application, the BS can obtain this information from the received packets in the startup phase of the sensor network.

The number of mobile intruders we choose in the experiments varies from 5 to 30 in step sizes of 5. These mobile intruders are randomly deployed in the initialization phase of the sensor network. Each mobile intruder picks a random direction to move at the speed of 5 units per second. Figure 12 shows the experiment results for detecting both static and mobile intruders. The results show that mobile intruders reduce the accuracy of the detection.

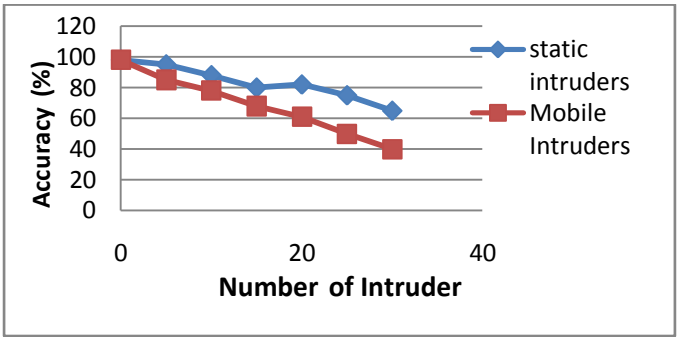


Figure 12. DC-inspired algorithm detecting static and mobile intruders

Figure 13 shows the simulation results of our dual protection scheme for a WSN. It is clear that the BS can obtain more accurate packets by using the dual protection scheme than by only performing the standard DC-inspired algorithm. The dual scheme effectively reduces the impact of the mobile intruders on the DC-inspired algorithm, and this advantage is more obvious when there are more mobile intruders in the network. When there are more mobile intruders in the sensor network, the chosen monitored point is more likely to be surrounded by the intruders. The BS estimates the impact of these intruders on the monitored point and then sends out the assessment to the sensor nodes around the monitored point. These sensors nodes effectively adjust their monitoring period according to the received assessment. This mechanism improves the detection rate to some extent.

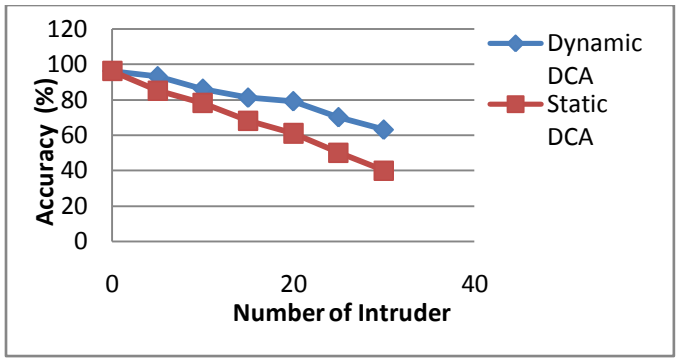


Figure 13. Dynamic DCA vs static DCA on detecting mobile intruders

In summary, our DC-inspired algorithm effectively detects static intruders in a WSN, but mobile intruders can elude detection. The MCMC technique and the fuzzy logic algorithm lower the effect of the mobile intruders and render the algorithm suitable for WSNs with either static or mobile intruders.

VI. CONCLUSION AND FUTURE WORK

We have presented a dual security mechanism for WSNs. A sensor node performing the dendritic cell-inspired algorithm effectively detects harmful intruders that report fake messages, transmit changed packets or report messages with unexpected frequencies. To enhance the ability of the dendritic cell algorithm to detect mobile harmful intruders, we developed an impact assessment system. The enhanced system uses the Metropolis-Hastings algorithm to infer the location of intruders based on partial information, followed by a fuzzy logic algorithm for assessing the impact of intruders on a monitored point. The assessment directs a sensor node to dynamically adjust the monitoring period according to the current network situation. Our simulations demonstrate that the dual protection mechanism promptly and effectively detects static or mobile intruders, extends sensor network lifetime, and improves the accuracy of the packets received by the BS.

In many real applications, such as battlefield monitoring, the number of intruders in a monitoring area will dynamically change. To apply to this kind of application, our security system would need an enhanced MCMC algorithm for tracking multiple moving targets. The question of scheduling feedback from the BS to control feedback for the monitoring period is another open question for investigation.

REFERENCES

[1] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, and S. Hailes, “Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm,” Proc. International Conference on Artificial Immune Systems (ICARIS), pp. 390–403, 2006.

[2] L.N. DeCastro and J. Timmis, "Artificial ImmuneSystems: A New Computational Intelligence Approach," Springer-Verlag, 2002.

[3] U. Aickelin and J. Greensmith, “Sensing Danger: Innate Immunology for Intrusion Detection,” Elsevier Information Security Technical Report, pp. 218–227, 2007.

[4] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod, “Danger theory: The link between AIS and IDS,” Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS 2003), vol. 2787 of LNCS, Springer-Verlag; pp. 147‑155, 2003.

 ISBN: 978-1-61208-107-6

[5] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection," Proc. International Conference on Artificial Immune Systems (ICARIS), pp. 153–167, 2005.

[6] N. Mazhar and M. Farooq, "A sense of danger: dendritic cells inspired artificial immune system for MANET security," GECCO, pp. 63-70, 2008.

[7] D.B. Johnson and D.A. Maltz, "Dynamic source routing in adhoc wireless networks," T. Imielinski and H. Korth, eds., Mobile computing (Kluwer Academic) chapter5, pp. 153-181, 1996.

[8] J. Greensmith, U. Aickelin, and J. Twycross, "Articulation and Clarification of the Dendritic Cell Algorithm," ICARIS, LNCS 4163, pp. 404-417, 2006.

[9] A.P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks (Q2SWinet '05). ACM Press, pp. 16–23, October 2005.

[10] M. Nikraz, G. Caire, and P. A. Bahria, "A methodology for the analysis and design of multi-agent systems using jade," International Journal of Computer Systems Science and Engineering, vol. 21, no. 2, 2006.

[11] M. Vinyals, J. A. Rodr ́guez-Aguilar, and J. Cerquides, "A Survey on Sensor Networks from a Multi-Agent perspective," 2th International Workshop on Agent Technology for Sensor Networks (ATSN), 2008.

[12] R. Biswas, S. Thrun, and L.J. Guibas, "A probabilistic approach to inference with limited information in sensor networks," in: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN' 04), Berkeley, CA, pp. 269‑276, Apr. 2004.

[13] J. Kim and P.J. Bentley, "Evaluating Negative Selection in an Artificial Immune System for Network Intrusion Detection," Proc. Genetic and Evolutionary Computation Conference (GECCO), pp. 1330-1337, 2001.

[14] S. Forrest, S. Hofmeyr, and A. Somayaji, "Computer Immunology," Communications of the ACM, 40(10), pp. 88-96, 1997

 ISBN: 978-1-61208-107-6