

Danger Theory and Multi-agents Applied for Addressing the Deny of Service Detection Problem in IEEE 802.11 Networks

Moisés Danziger, Marcelo Lacerda and Fernando B. de Lima Neto, *Senior Member IEEE*
Department of Computing and Systems - Polytechnic School of Pernambuco
University of Pernambuco, Recife-PE, Brazil.
E-mails: {md, mgpl, fbln}@dsc.upe.br

Abstract

Deny of service (DoS) detection problem is a common and annoying network difficulty, but for IEEE 802.11 standards it becomes even more troublesome. Addressing this issue, we introduce a new approach to promptly warn the user. The detection algorithm put forward, combines second generation of Artificial Immune Systems, Danger Theory and Multi-Agent System. For the detection system, we used the dendritic cells algorithm, modified to IEEE 802.11 environments. Experimental results carried out in controlled setups have shown that the model can easily and effectively be applied for detecting DoS in IEEE 802.11 networks.

1. Introduction

Intrusion detection systems (IDS) are well known and used for monitoring of networks and guarding computers. They constitute a substantial part of any computer security architecture. However, there are some limitations that can generate vulnerabilities; inability to detect new types of attacks is one example. Inspired by human biology and taking into consideration the similarities regarding protection abilities, artificial immune systems (AIS) were adapted to reduce the impact of new types of intrusion [1][2].

The negative selection algorithm (NSA) was one of the most used implementation models for detection, but, it is necessary to map what is self-nonSelf (SNS) in the system or network; this is not easy, especially regarding issues of scalability and false positives [1].

In 2003, Aickelin et al [2], proposed a new model of IDS based on the danger theory (DT) of Matzinger [3]. According to DT, AIS is then activated through the analysis of danger signals issued by dendritic cells (DC). Based on DT, the SNS task is replaced by the

analysis of the concentration of different signals from the environment where DCs were exposed.

In biology, DCs act as the interface between the innate system and the adaptive system of AIS. The DCs are able to process different types of signals and produce their own signals in a process known as signal transduction. After processing, the DCs present the antigens (here, source of the signals collected) in order for the system to adapt to a given context. Then, according to the antigen context, the AIS can respond adequately to the problem activating T-cells.

A modified version of AIS is also presented in this paper. We use a model adapted from the dendritic cells algorithm (DCA) developed by Greensmith [4]. The DCA is based on an abstract model of the behavior of DCs, successful in detecting scanning anomalous activities in networks [5] [6].

The main difficulty to implement a model based on DT is how to define, explore and sense danger. The majority of current danger models are still unable for dealing with the real network traffic [8]. In the literature there are some patterns of scan behaviors used to create danger signals in network [5]; patterns of danger for *Bot* detection are also available [7].

Most of the models developed using the DT to detection are used in structured environment (i.e. cables). Conversely, in this paper we focus on the detection deny of service (DoS) in IEEE 802.11 networks. This decision is motivated by the current extreme vulnerability found in this medium.

To verify the feasibility and capacity to function in a network model based on the IEEE 802.11 standard, we propose in this paper hierarchical multi-agents AIS, based on DT to carry out intrusion detection. Initially, two types of agents were used. Each one of them has distinct functions, similar to those found in the model based on DT, namely, (i) detection by dendritic cells, (ii) innate response by memory T-cells, (iii) adaptive response by creating antibodies to unknown intruders and (iv) removal of cells to prevent instability of the

system. Our main aim and thought contribution is test the new AIS approach in DoS attacks on IEEE 802.11.

This paper is organized as follows: sections 2-4 contains a brief review of related works and concepts about the three main areas involved in this work. In section 5 we provide the put forward model design. In section 6 details of modeling and construction of our approach are commented upon. In section 7 simulations and results are presented.

2. Danger theory, dendritic cells and DCA

2.1 Danger theory

Since 1960s, when the SNS theory was proposed, many works have been developed using the concepts of negative and positive selection for security systems. However, in almost all of them, the researchers pointed out two problems: i) large number of false positives and ii) difficulties with the system size. These problems hindered the application of those models in real world environment (i.e. in real-time). Thus, another approach was needed for the AIS in this domain.

The DT suggests that in the context of human immune system (HIS) there is a different point of visualization to decide when the body suffers any threat. In most cases, the tolerance for potential threats is great and the AIS does not need to combat all found non-Self in the body. According to Matzinger, DT does not depart from SNS principles, but offers a solution to prevent the collapse of the system in trying to eliminate everything that is alien. The whole process occurs because the attacker, according to DT, may stimulate the generation of cellular molecules (danger signal) by initiating cellular stress or cell death [9]. There are two classes of danger signals, generated: (i) by the body itself and (ii) by invading organisms.

The interest on DT for IDS is the differentiation of cell death, in this case necrotic (“bad”) and apoptotic (“good” or “planned”), which are presented by the antigen presenting cells (APCs) for activation of the IS [2]. The apoptosis has a suppressive effect while necrosis a stimulatory immunological effect and cannot be as distinct. The correlation of these two effects is the base of danger signals to be used in IDS context.

2.2. Dendritic cells

There are two distinct systems in IS: (i) the innate system and (ii) the adaptive immune system. The innate is the first line of defense against attacks. The cells of innate system have the ability to detect and dispose of pathogens through phagocytosis. The innate

system provides information on all previous contact with foreign invaders than those coming from genes.

The DCs belong to the innate system and are a family of cells known as macrophages. The main function of DC is to clear the tissue of debris. There are three different states for DC: (i) immature, (ii) semi-mature and (iii) mature. Initially, each DC is in the immature state and according to the concentration of signals received, their states can change [10].

In immature state, DC collects debris and some of them are used as antigen. Antigens are processed by DCs through the receptors expressed in the cell surfaces, which sense the various signals. After that processing, the DC calculates the potency and concentration of signals eventually may change their own states.

There are three signals that DCs are sensitive of: (i) pathogen associated molecular patterns – PAMPs, (ii) danger signals – DS and (iii) safe signals – SS. PAMPs are exogenous signals produced exclusively by pathogens (i.e. they represent anomaly situation). The DS can be a cells responding to micro-organisms or derived from dying cells. This sign may or may not indicate an anomaly situation. In the case of SS, the sign is safe – e.g. normal cell death, and indicate normality of system. When the DCs are exposes to these signals, they cause the production of molecules called cytokines, which can activate or suppress the IS.

To the signal transduction process, the DC has to be exposed to greater quantities of signals. If most signals, either PAMPs or DS, present sum of greater than the SS, this cause the DC to changes into the mature state. Next, the DC migrates from the tissue to the lymph node and activates the T-cells through the inflammatory cytokine called Interleukin-12. In the lymph node, DCs present the antigens and, to facilitate this process, they produce the co-stimulatory molecules (CSM).

Accumulation of SS will help to transform more cells into semi-mature DC. The difference to mature DC is that in this case, they do not have ability to activate T-cells. The semi-mature DC produces Inteleuakin-10 that suppresses T-cells through inhibition the production of Interleukin-12 while increasing production of Interleukin-10. In this state, DC present antigen in that may lead, without control, for possible problem and help in prevention of autoimmunity [11].

In summary, DC has the power to control the adaptive immune response [12] and [13].

2.3. The Dendritic Cell Algorithm

Proposed by Greensmith *et al.* and first introduced in 2005 [14], the DCA is an algorithm belonging to the

“2nd Generation” of AIS. The algorithm uses principles from DT to perform contextual intrusion detection and has been used to some other security problems as port scans, Bot detection, and as a basis for new IDS models. An important result found in Greensmith work’s is the application for real-time problems. This is a huge shortcoming for “1st Generation AISs”.

The DCA is a population based system. The cell (i.e. representative of the population), in this case is an abstraction of the DC in nature. Each cell can collect data items (i.e. antigens) and process the value of input signals. Figure 1 has details of DC structure.

There algorithm has two parts: (i) one for data collection – tissue and (ii) another for analysis of antigens - lymph node. In the first part, data is stored to be presented for DC population. There are three main data structures, namely: (a) antigen array, (b) signal matrix and (c) array of objects (i.e. each DC is an object). In the second part of DCA, after processing all input signals, the process of analysis is triggered to verify the context of antigens. The change of DC states, previously explained, is governed by Equation 1 and Table I.

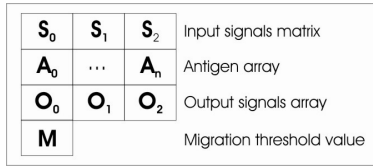


Figure 1. Dendritic Cell components

$$o_p(m) = \frac{\sum_i \sum_{j \neq 3} W_{ijp} * S_{ij}(m)}{\sum_i \sum_{j \neq 3} |W_{ijp}|} \quad \forall p \quad (1)$$

where:

- W is the signal of category i
- S is the tissue signal matrix
- i is the input signal category
($i_0 = \text{PAMP}$, $i_1 = \text{DS}$, and $i_2 = \text{SS}$)
- j is the output signal value
($j_0 = \text{CSM}$, $j_1 = \text{semi-mature}$, $j_2 = \text{mature}$)

Table 1. Signal Weight Values

W_{ijp}	$j = 0$	$j = 1$	$j = 2$
$p = 1$	2	1	2
$p = 2$	0	0	3
$p = 3$	2	1	-3

The DCA has three main stages: initialization, update and aggregation. In the initialization happens all necessary setting of various parameters (e.g. number of DCs in population, number of input signals per category, number of cycles, and others). The update stage is a continuous process and the value for signals

and antigens refreshed upon the arrival of new data. In this paper, we used one cell cycle per second. The cell cycle defines a discrete process in the algorithm, but for real-time application it will be necessary a continuous mode.

Once one DC reaches the migration threshold (i.e. through CSM accumulation), this DC is removed from the population and, the content of signal array and values context are logged to a file for the aggregation stage. This stage does not occur until all data are processed by DCs (i.e. in discrete case). The main function to this stage is to calculate the mature context antigen value (MCAV) that is used to assess the degree of anomaly of a give antigen. The MCAV is shown in Equation 2. The range of action is between 0 and 1; the closer to 1 means the most anomalous antigen.

$$MCAV_x = \frac{Z_x}{Y_x} \quad (2)$$

$MCAV$ is the coefficient for antigen type x ; Z_x is the number of mature context antigen presentations for antigen type x ; and, Y_x is the total number of antigen presented for antigen type x .

For implementation of the DCA, we used an adaptive model based on version [11]. This is due to the fact that we did not use the *libtissue* project [15], but our new implemented model in Java language.

3. Multi-agent system

Multi-agent system (MAS) has been a great open field for research with applications in many areas of knowledge. Moreover, a great deal of research involving artificial intelligence (AI) and MAS has been produced and delivered in the recent years. Multi-agents can be defined as a collection of computational entities that have ability to solve problems and can act in a cooperative or individual manner through the exchange of information. It is intuitive that an isolated agent is less likely to solve a distributed task than a collection of them.

In this work, the hole of MAS is related to the distributed aspect of detection will be used in the system.

3.1. Agent

According to Russel and Norvig [16], agents use sensors to perceive their environment and actuators to act upon it. Agents can be of different types and have some sophisticated features to increase their performance in the environment which is inserted. However, the core of the agent will always be [17]: (i) its perception of the environment, (ii) its ability to

interact with other agents and (iii) its ability to initiate and persistently pursues its own goals.

Embedded in the agents “mind” some intelligent engine can afford them with adaptability, which is an important feature in dynamic environments. Intelligence is also quite interesting to distributed systems such as IDS [18]. In this work the AIS based on DT is the intelligent engine; this will be explained later in the design model section.

4. Intrusion detection on IEEE 802.11 networks

The IDS has its origin in the wired infrastructure network, and its use is as old as computer viruses. Two action form the basis of any IDS: (i) based on evidence of intrusion and (ii) based on the deviation of behavior. These two actions can be called misuse-based IDS for ‘i’ and anomaly-based IDS for ‘ii’. Here we follow the second one.

Some approaches found for IEEE 802.11 intrusion detection are: MAS model [19]; model based on multi-channel monitoring & anomaly analysis using adaptive machine learning and genetic search [20]; works with sequence number-based MAC address spoof detection [21]; and, specification-based approach [22]. In this paper we focus on the anomaly intrusion method together with a new approach based on AIS to verify the application of DT.

4.1. Security in IEEE 802.11 networks

The technique used in this paper achieve in the first instance only to the IEEE 802.11 infrastructure network. This model has a central agent termed Access Point (AP) that manages the communication between the nodes. The main vulnerabilities are situated exactly on the management carried out by AP, i.e. the authentication and association between user and AP.

The exchange of messages is done through MAC layer management protocols and the latest version of the standard, the IEEE 802.11i, has brought many improvements to the problems of these protocols, but did not solved a serious problem to reduce the chances of an effective and disturbing type of attack in IEEE 802.11 networks: the DoS attacks. This weakness is caused by the lack of authentication of management frames, which allows any networks IEEE 802.11 (i.e. the IEEE 802.11i too) are vulnerable to spoofing types attacks [23].

According to previews works [24] and [25], even with the IEEE 802.11i and the robust security network association (RSNA) which allows mutual authentication, introduces key management protocols

and new data encryption and integrity protocols, the management frames used to authentication by extensible authentication protocol (EAP) are unprotected. So, there are possible chances to intruders planning attacks with these frames.

The IEEE 802.11 network attacks can be classified into four categories: identity spoofing, eavesdropping, vulnerability, and flood attacks (Anomaly-Based). It is also possible to classify the attacks into passive or active. The passive mode happens when the attacker only collects data from the network (e.g. using a sniffer tool) to know and possibly plan the attack assets. In the case of active mode, the attacker needs to send some information into the network, (e.g. a managed frame flood). We used the passive and active mode to the experiments as will be explain in experiments section.

4.2. DoS for IEEE 802.11

Denial of service is an old known network problem and IEEE 802.11 networks are much desired by the malicious DoS attackers [20]. They presented two methods with learn capacity to IEEE 802.11 intrusion detection: one based on in hidden Markov model (HMM), another one based on adaptive resonance theory. In [23] is possible to find DoS attacks types with emphasis on the routing layer and MAC layer, and possible methods to defense. There are two some modifications for IEEE 802.11 standards improve protection of network equipments against MAC layer DoS attacks. However, as seen above, there are problems that can undermine the use of the network through the unavailability caused by a successful DoS attack.

The deauthentication broadcast flood is a known model of DoS attacks and its use can leave the network simply unworkable. This type of attack occurs when an attacker can send in continuous or discrete deauthentication frames (i.e. managed type) from AP to station or station to AP. When the attacker is successful, the capacity of network the transmission is reduced.

5. Design of immune-based on DT-MAS

For presenting a new approach of intrusion detection, this paper has designed a model to detect anomalies using agents with an immune-inspired intelligence engine, DT-based. This system includes two types of agents: (i) basic agents and (ii) subaltern agents. To develop the agents, we use the Java Agent Development Framework (JADE)[26]. This framework is open-source, allowing customization and has a hassle-free development environment.

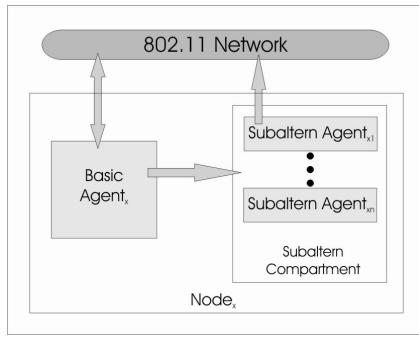


Figure 2. Model of AIS-DT of MAS.

5.1. Model of the AIS-DT agent

As commented in the section 2, the DT was used on AIS through DCA and T-cells features. The structure of the agent used here can be seen in Figure 2.

5.1.1. Basic Agent

The main agent of system is responsible for detecting signs of failure through application of the DT-concepts. This agent contains the modified DCA algorithm. Figure 3 shows modules of the Basic agent; notice that the packets analyzer, an important process, is placed before data presentations for DCA. In this module, the network packets were processed and separated generating antigens with respective raw input signals.

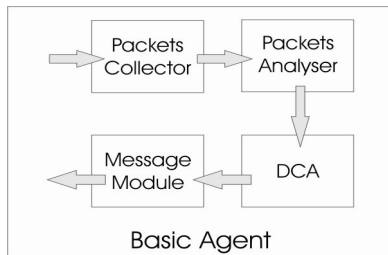


Figure 3. Structure of Basic Agent

In the subsection signs and antigens we describe the context to collect the signs and for antigens. Initially, the basic agent is only tested locally (i.e. we prepared agents to nodes, but only tested in one node).

5.1.1. Subaltern Agent

This agent mimics the T-cells functions and can have various types, depending on attack type. When the basic agent defines antigens with regard danger, it sends a message to the subaltern part of the algorithm with embedded antigens. After receiving this message, the antigens are presented to the subaltern agents and

through a process defined as affinity; they try to identify what type of attack or danger it belong to. There may be a situation where the type of danger is not recognized by subaltern agents; in this case, it will send a message to another specialized agent, which for the moment is not yet implemented.

To facilitate the process of affinity, the antigens are created with a structure which defines some basic characteristics (e.g. packets source, packets destination, packets size, base station, etc.). Similarly, when agents are created, one or more structures are defined and may generate some affinity with the antigen.

Besides the agent structure array, there is the memory array, which records the entire history of agent activity (i.e. the more history it have, more active is the agent and this may be a more problems). Figure 4 shows the data structure of subaltern agent. Each subaltern agent has a response model array that store types of action relative to attack. This response is defined based on the concept of degree of risk [27].

F_0	...	F_n	Structure array
M_0	...	M_n	Memory array
T_0	...	T_n	Types of response array

Figure 4. Structure of Subaltern Agent

6. Methodology

6.1. Attack scenarios

For the purpose of experimentation we have built two types of scenarios: (i) easy scenario and (ii) complex scenario. The main idea is checking the level of detection for both cases, verifying the rate of false positives and false negatives for each.

In the easy scenario, we used three computers and one IEEE 802.11g antenna. To verify the capability of detection to the DCA, we have simulated a DoS attack with deauthentication packets insertion using the Aircrack-ng [28] by spoof MAC address on a station. The complex scenario uses the same attack on an environment in where several computers were transmitting data through IEEE 802.11g antenna.

The packets collected during simulation were stored in a *.dump* file. We used the Wireshark [29] tool to save data in the text file. To simulate a normal process in the network, which generates high traffic, we use a FTP transfer between two stations on the network. In a normal case, the network present a high rate of packets per second, but at the moment the attacks occurs, the rate of packets per second decreased significantly, reaching zero at given moments.

Important to notice that in certain situations the rate of packets per second was low but this did not mean an attack. To resolve this problem, we study some important signals that can show the abnormal behavior. For more explanation about signals refer to the signals and antigens section, below.

6.2. Signals

Signals are mapped directly from packets analyzes. Three signal categories are used to define the state of the system. These signals are collected using the implemented packets analyzer embedded in the basic agent. When the signals are collected, they are in different format and therefore require a normalization process. We use the normalization for the three signs through the normalization function well known to these problems as see in Equation 3. To PAMP signal, we used the rate of small packets per second. After analyzing the packets database, it was verified that the packets of the deauthentication type were between 10 and 30 bytes. But, there were other types of packets in this range (e.g. the acknowledgment packet), so, to facilitate the process we use a range greater than 10 and less than 30 with maximum value on scale; for other sizes, we applied 0.

The danger signal is derived from the number of transmitted network packets per second. It can be high or low depends on network traffic, but in normal transfer, the transmission rate is high. In contrast, when a deauthentication attack happens, the packets rate drops at very low levels. This sharp decline may also mean the end of transmission (e.g. of a large file) which cannot be considered a threat, but a normal situation. For this, analysis was made of the highest peak in the rate of transmission and standard range through normalization.

Finally, the safe signal is derived from the range of types of packets variation per second. In a transmission of packets (e.g. via FTP) the standard rate of change is almost zero, the opposite occurs when there is a case of attack or danger. For this signal was also performed the same scale normalization from two previous signals.

To observation, is important to notice that we work directly with the packets and not with system calls functions as other works. This involved the creation of antigens and the distribution of signals. The fusion of these three signals is shown in Figures 5 and 6.

$$V = \frac{V - V_{\min}}{V_{\max} - V_{\min}} * 100 \quad (3)$$

6.3. Antigens

For these experiments we used each packet as an antigens (i.e. the packet generate an antigen). At this stage, the antigen loads in its structure the main characteristics of packets. This will allow the verification of affinity between the antigens have set for the attack or danger to the subaltern agents. In fact, little processing is done on the antigen at this stage, because the detection only works with antigen and correlated input signals. By processing of antigen, this refers to the second by which antigen is collected and presented for analysis by the DCs. For this process it is important that the actual value of antigen not to be changed as it may compromise the output signals.

6.4. System setup

All experiments were performed on an AMD Turion/64 X2 1.98GHz running Ubuntu Linux machine (kernel 2.6.26). The DCA algorithm is implemented within the JAVA languages and for agents we used the JADE framework. All packets are collected from Aircrack-ng tool and subsequent transformed in text file with Wireshark tool. The text file is the data base of packet analyzer. The default parameter settings are in Table 2.

Table 2. Settings and Results of Experiments

Experiment	Easy	Complex
Packets Size	293923	299324
Antigens	294245	18292
Migration Threshold	1-2	1-2
DC Population	296	934
Mature Cell	11	741
Semi-Mature Cell	285	193
MCAV	0.0372	0.79336188
False Positive	6	2
False Negative	0	1
Error Rate	2,03%	0,32%

7. Results and Analysis

The results of the experiments are shown in the figures below, in Figure 5 - the easy scenario and in Figure 6 - the complex scenario. For the easy scenario, the attack occurred during each 3 minutes of FTP transmission.

It can be noticed that the DS signal is almost zero while the PAMP signal has two peaks. The SS signal slumped at the same instant of PAMP act. We used 100 as the maximum of secure SS signal.

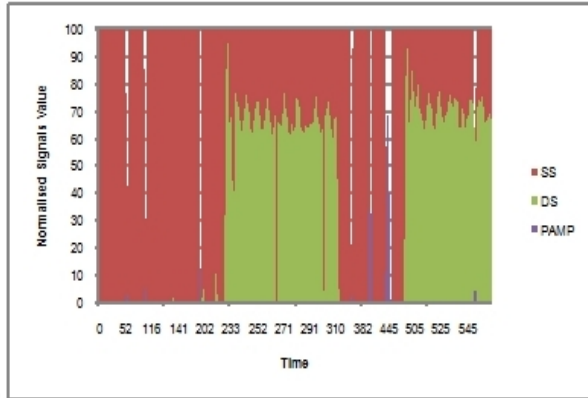


Figure 5. Sample of signals during DoS attack on FTP transmission.

In the Figure 6 the signals are more variable than in the first; this may be caused by the greater network traffic. As shown, several attacks have occurred during the experiment and after these attacks; there were perturbations on the variation of packets type and size. So, the network was not “at easy”.

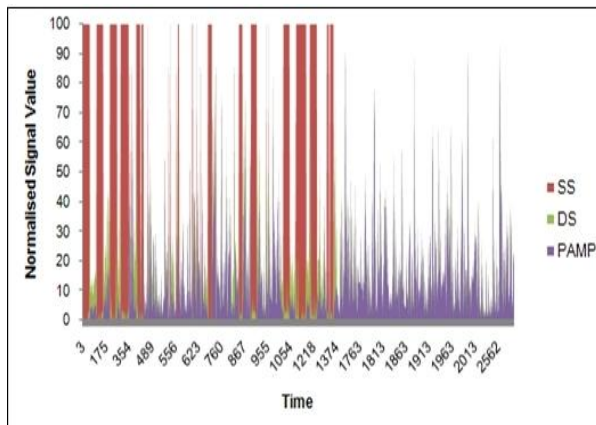


Figure 6. Sample of signals during DoS attack on mixed types of processes

We observed that, among the apparent disorder, there was a tendency in the behavior of network traffic. The peculiar characteristics of IEEE 802.11 networks force the authentication and association. This process generates many changes of manage packets. When a *deauthentication* attack occurs, for example, the station needs to connect again and this takes some time. So, even if an attack occurred during a few seconds earlier, the algorithm was able to detect those posterior seconds with bad behavior. This ability helped in reducing the false positive and false negative as shown Table 2, especially in the complex scenario.

The false positive figure from easy scenario is greater than complex scenario. We verified that this occurs because the number of packets per second,

when there is a file transfer, can be very large. This contrasts with a normal flow of packets in the network (e.g. for sample experiment: maximum = 2333 and minimum = 1). This high contrast can harm the cell, hence normalization is needed. For MCAV there is a gap between easy scenario and complex scenario. This is expected because MCAV defines the ability to detect the cell. In Table 2 shows relations between the antigen sizes, quantity of cells and number of mature cells. Results show that for the two cases the algorithm is able to detect intrusion. The migration threshold used a random number ranging between 1 and 2.

As for observation, we tested the algorithm with various DCs, but, all the antigens were presented to cell once. So, all values become equal when applied to multiple cells. Thus, in this work we decided to use one cell alone and the results were quite good.

8. Conclusions and future work

DCs are interesting inspiration to use in various problems. Through their peculiar signal processing abilities, they can detect anomalous behaviors even in complex environments. In this paper we applied DCA adapted for IEEE 802.11 networks where the antigens are the packets. However, to work on these abundant antigens, it was necessary to apply input signals at every second, which hampered implementations of several cells. But, even with only one cell, the results encountered for DoS problems were deemed to be satisfactory.

Notice that we use only two types of agents, emphasizing that the basic agent is responsible for DCA detection. For future work, we thought of three different research directions, namely: (i) to compare the performance of the present model with other AIS based model for anomalous detection and DoS attacks, (ii) to study and implement a model that support several cells per second, and (iii) to improve the performance of the current model by “recruiting” other types of agents more elaborate than the subaltern agent. Now, the subaltern agent only receives detected antigens and defines the type of attack.

9. Acknowledgments

The authors thank to Polytechnic School – University of Pernambuco, FACEPE and CNPq (Brazil).

10. References

[1] T. Stibor, P. Mohr, J. Timmis, and C. Eckert, “Is negative selection appropriate for anomaly detection?”, In *Proceedings of Genetic and Evolutionary Computation*

- Conference (GEECO), Washington DC. USA. 2005, pp. 321-328.
- [2] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod, "Danger Theory: The link between AIS and IDS", In *Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS-03)*, Edinburgh, UK., September 1-3, 2005, pp. 147-155.
- [3] P. Matzinger, "Tolerance, danger and the extended family", In *Annual Reviews in Immunology*, 12, pp. 991-1045
- [4] J. Greensmith, "The Dendritic Cell Algorithm", *PhD Thesis*, University Of Nottingham, 2007.
- [5] J. Greensmith and U. Aickelin, "Dendritic cells for SYN scan detection", *Proceedings of the IEEE Genetic and Evolutionary Computation Conference (GECCO-07)*, London, UK. 2007, pp. 49-56.
- [6] J. Greensmith, J. Twicross, and U. Aickelin, "Dendritic cells for anomaly detection", In *IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 16-21, 2006, pp. 664-671.
- [7] Y. Al-Hammadi, U. Aickelin, and J. Greensmith, "DCA for bot detection", In *IEEE Congress on Evolutionary Computation*, Hong Kong, June 1-6, 2008, pp. 1807-1816.
- [8] H. Fu, X. Yuan, K. and N. Wang, "Multi-agents artificial immune system (MAAIS) inspired by danger theory for anomaly detection", In *IEEE International Conference on Computational Intelligence and Security Workshops*, Harbin, Heilongjia, China, December 15-19, 2007, pp. 570-573.
- [9] P. Matzinger, "The Danger Model: a renewed sense of self", *Science*, 296 (12 April 2002) 2002, pp. 301-305.
- [10] M. Lutz, and G. Schuler, "Immature, semi-mature, and fully mature dendritic cells: which signals induce tolerance or immunity?", In *Trends in immunology*, 23(9):991-1045, 2002
- [11] J. Greensmith, U. Aickelin, and G. Tedesco, "Information fusion for anomaly detection with the dendritic cell algorithm", accepted for the Special Issue on Biologically Inspired Information Fusion, to be appear in *International Journal of Information Fusion*, Elsevier, 2007.
- [12] C. A. Janeway, "Approaching the asymptote? Evolution and revolution in immunology", *Cold Spring Harb Symp Quant Biol*, 1989, pp. 1-13.
- [13] T. R. Mosmann and, A.M. Livingstone, "Dendritic Cells: the immune information management experts", *Nature Immunology*, 2004, pp. 564-566.
- [14] J. Greensmith and U. Aickelin, and S. Cayzer, "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection.", In *Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS-05)*, LNCS 4163, 2006, pp. 404-417.
- [15] J. Twycross and U. Aickelin. "Libtissue – implementing innate immunity.", In *Congress on Evolutionary Computation (CEC-2006)*, LNCS 4163, 2006, pp. 499-506.
- [16] S. Russel and P. Norvig, *Artificial Intelligence: a modern approach*, Prentice Hall, 1995.
- [17] R. Zhang, D. Qian, C. Bao, W. Wu, X. Guo. *Multi-agent based intrusion detection architecture*, 2001. pp. 494-501
- [18] H. Fu, X. Yuan, K. Zhang, X. Zhang, Q. Xie, "Investigating novel immune-inspired multi-agent system for anomaly detection", In *IEEE Asia-Pacific Services Computing Conference*, 2007, pp. 466-472.
- [19] W. Hairui and W. Hua, "Research and design of multi-agent based intrusion detection system on wireless network", In *International Symposium on Computational Intelligence and Design*, 2008. pp. 444-447.
- [20] D. Tian and Q. Li, and S. Chen "Anomaly Intrusion Detection methods for wireless LAN", In *Fourth International Conference on Natural Computation*, 2008. pp. 179-182.
- [21] F. Guo and T. Chiueh, "Sequence number-based MAC address spoof detection.", In A. Valdes and D. Zamboni editors, *RAID*, volume 3858 of LNCS, Springer, 2005 pp. 309-329.
- [22] R. Gill, J. Smith, and A. Clark "Specification-based intrusion detection in WLANs", In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06)*, 2006. 10p.
- [23] J. Bellardo, and J. Savagi, "802.11 Denial-of-service attacks: real vulnerabilities and practical solutions", In <http://www.cs.ucsd.edu/users/savage/papers/UsenixSec03.pdf> (Accessed in June of 2009).
- [24] A. Mishra and W. Arbaugh, "An Initial Security analysis of the 802.1x standard", *Technical report*, CS-TR-4328, UMIACS-TR-2002-10, University of Maryland, College Park, MD, 2002. At <http://www.cs.umd.edu/~waa/1x.pdf> (Accessed in June of 2009).
- [25] C. He and J. C. Mitchel, "Security analysis and improvements for IEEE 802.11i.", In *Proceedings of the 12th Annual Network and Distributed System security Symposium*, 2005.
- [26] JADE – *Java Agent Development Framework*, At <http://jade.tilab.com/> (Accessed in June of 2009).
- [27] J. Cacheand, and V. Liu, *Hacking exposed wireless: wireless security secrets & solutions*, Mc-Graw Hill/Osborne, 2007. 416 p.
- [28] Aircrack-ng – *802.11 WEP and WPA-PSK keys cracking program*, At <http://www.aircrack-ng.org/doku.php> (Accessed in June of 2009).
- [29] Wireshark – *A network protocol analyzer*, At <http://www.wireshark.org/> (Accessed in June of 2009).