A Sense of Danger: Dendritic Cells Inspired Artificial Immune System (AIS) for MANET Security

Nauman Mazhar
Deptt of Electrical and Computer Engg
Michigan State University
MI 48823, USA
naumaz@msu.edu

Muddassar Farooq
nexGIN RC,
NUCES-FAST,
Islamabad, 44000, Pakistan
muddassar.farooq@nu.edu.pk

ABSTRACT

AIS-based anomaly detection systems classically utilize the paradigm of self/non-self discrimination. In this approach, an algorithm learns self during a learning phase, therefore, such algorithms do not have the ability to cope with scenarios in which self is continuously changing with time. This situation is encountered once malicious nodes are to be detected in a Mobile Ad Hoc Network (MANET). Consequently, it becomes a challenge to differentiate a valid route change due to mobility from an illegal one due to tampering of routing information by malicious nodes. In this paper, we propose a dendritic cell based distributed misbehavior detection system, *BeeAIS-DC*, for a Bio/Nature inspired MANET routing protocol, *BeeAdHoc*. Our proposed system inspires from the danger theory and models the function and behavior of dendritic cells to detect the presence or absence of danger and provides a tolerogenic or immunogenic response. The proposed detection system is implemented in a well-known ns-2 simulator. Our results indicate that our detection system not only enables *BeeAIS-DC* to dynamically adapt its detector set to cater for a changing self due to mobility of nodes, but also is robust enough to provide significantly smaller *false positives* as compared to self/non-self based AIS. Moreover, the danger theory related overhead of *BeeAIS-DC* is minimal, and as a result, it does not degrade traditional performance metrics of *BeeAdHoc*. This behavior is vital for battery/bandwidth constrained mobile nodes.

Categories and Subject Descriptors

C.2.0 [**General**]: [Data communications, Security and protection (e.g., firewalls)]; C.2.2 [**Network Protocols**]: [Routing protocols, Protocol verification]

General Terms

Algorithms, Design, Performance, Security

Keywords

Artificial Immune Systems, Dendritic Cells, Misbehavior Detection, Mobile Ad Hoc Networks, Self Organization

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION

Artificial Immune Systems (AIS) are inspired from the working of Biological Immune System (BIS) [5]. They have been extensively studied to protect a computer system against intrusions by attackers in general and network anomaly detection in particular. The authors of [3] provide a comprehensive review of application of AIS to network anomaly detection.

Recently, Mobile Ad Hoc Networks (MANETs) is becoming an active area of research. The classical reactive routing protocols for MANETs are: *DSR* (Dynamic Source Routing) [9] and *AODV* (Ad-Hoc On-demand Distance Vector Routing) [13]. Similarly research in Bio/Nature routing protocols has resulted in state-of-the-art protocols like *AntHocNet* [4], *BeeAdHoc* [17] and *Termite* [15]. An important focus of research is now on understanding the impact of misbehaving nodes in a MANET environment. The security provision in MANETs is a challenge because wireless medium is inherently insecure. All nodes in the transmission range of a node can overhear its transmissions and at the same time initiate spurious transmissions of their own. Therefore, MANETs provide malicious nodes an ideal environment for fabricating and launching different types of routing attacks. As a result, they can not only disrupt the routing behavior of a routing protocol but also significantly degrade the network performance [12].

A number of security solutions have been proposed for MANET routing protocols based on either standard cryptography or classical self/non-self AIS. The classical security solutions are: (1) *ARIADNE* [8] that utilizes symmetric cryptography to secure the *DSR* protocol, and (2) Secure Ad-Hoc On-demand Distance Vector *(SAODV)* [19] that uses asymmetric cryptography for security of *AODV*. Similarly, in the domain of nature inspired MANET routing protocols, we studied the security vulnerabilities of *BeeAdHoc* protocol and proposed a security framework, *BeeSec* [12], based on digital signature authentication. We have shown that though the framework successfully counters a number of attacks, but it puts heavy communication and computation load on already bandwidth and energy constrained nodes.

AIS appears to be a promising paradigm to provide security in MANETs because of its low communication and computation overhead. The authors in [16] used AIS for misbehavior detection in a network running *DSR*. The system is able to detect dropping attacks launched by multiple nodes simultaneously. We proposed an AIS security solution *BeeAIS* [11], that utilizes classical self/non-self paradigm to secure *BeeAdHoc*. Launching a number of fabrication and tampering attacks, we demonstrated that the *BeeAIS* protocol can successfully counter a number of routing attacks launched on *BeeAdHoc*. However, its communication and computational costs are not only significantly smaller as compared to *BeeSec*, but its performance metrics are also better than *BeeSec*. This pattern is inline with the results reported in [18], where the authors compared a cryptographic and an AIS based security framework for *BeeHive* protocol for fixed networks.

Sometimes self/non-self AISs use clonal selection to improve the secondary response of the system. Moreover they use negative selection to generate detectors's database at the end of the learning phase. Consequently, they are not scalable, have high false positives, suffer from inadequate coverage of detection space, and are not capable of adapting to the changing self. The last disadvantage renders this paradigm unfit for MANETs (due to its high false positives), where mobility of nodes results in frequent changes of routes between a source and a destination. In order to overcome this serious shortcoming, we propose in this paper *BeeAIS-DC* that takes inspiration from the *Danger Theory* paradigm. It utilizes the principles of the dendritic cells (DCs) to dynamically update the detector's set. It first detects the presence or absence of danger signals, and subsequently follows the dendritic cell differentiation pathways to model the new/changed self. The presence of a "danger context" is viewed as a malicious activity and vice versa. Since this context keeps on changing, therefore, it counters malicious nodes in a mobile network. To the best of our knowledge, this is the first attempt to provide DC based security in MANETs. We believe that the current work will pave the way for developing more robust, scalable and adaptive 2nd generation AIS.

Organization of paper. The rest of the paper is organized as follows. In Section-2 we introduce fundamental concepts of Danger Theory by emphasizing the function and behavior of the dendritic cells. In Section-3, we briefly describe the *BeeAIS* security framework already implemented for the *BeeAdHoc* protocol and then report its performance under mobility. We show that the performance of *BeeAIS* significantly degrades under mobility. We then introduce our proposed security framework, *BeeAIS-DC* in Section-4. In Section-5, we describe the attacker framework that is developed for ns-2. It launches a number of routing attacks on *BeeAIS-DC* and then we show that our proposed solution successfully counters the attacks. In order to verify that our security enhancements do not degrade the performance of the original *BeeAdHoc* algorithm, we compare *BeeAIS-DC* with *BeeAdHoc*, *AODV* and *DSR* protocols in Section-6. The results of our ns-2 simulations clearly indicate that the network performance of *BeeAIS-DC* is quite close to that of *BeeAdHoc* and even it achieves better performance compared to non secure version of *AODV* and *DSR*. Finally, we conclude the paper with an outlook to our future research.

2. DANGER THEORY

The *self/non-self discrimination* paradigm is a widely accepted viewpoint in immunology. The immunologists believe that adaptive immune system is activated once our body recognizes foreign entities (antigens). The second group of immunologists believe in Danger Theory [10]. The immunologists of this group believe that activation of the adaptive immune response requires the presence of "danger" in addition to the recognition of pathogen. "Danger" indicates damage to the body cells due to a pathogenic infection. This recognition is performed by Dendritic cells (DCs) of innate immune system. This effectively gives control of adaptive immune system to the innate immune system: it can suppress the response of adaptive immune system in the absence of "danger" in tissues.

Dendritic Cells (DCs). The DCs are Antigen Presenting Cells (APCs) that are responsible for sampling the antigens from the tissues, including self and non-self antigens, and then presenting these antigens in the *thymus* for *T-cells* maturity. Immune system cells, inclusive of DCs, communicate with each other through secretion of specific molecules, termed as "signals". When a body cell undergoes apoptosis (planned cell death), the signals generated are totally different once a necrotic cell dies because of pathogenic infection. The DCs are sensitive to relative concentrations of these signals in the fluid surrounding the cells in tissues. DCs express receptors on their surface that enable them to receive signals from the environment. DCs, therefore, act as information fusion agents. They receive information from different sources, process that information and then produce the appropriate *immunogenic* or *tolerogenic* response.

Depending upon the types of signals present in the residing tissue (*safe signals* or *danger signals*), the DCs may exist in one of the following three states:

Immature DCs. A DC initially arriving in the tissue is in *immature* state. In this form, it acts as a *phagocyte* to clear the tissue of cell debris, and also collects antigens, presenting them on the cell surface. An *immature* DC, depending upon types of the signals present in the tissue and their relative concentrations, transforms itself to a *semi-mature* or the *mature* state. A higher concentration of PAMPS (Pathogen Associated Molecular Patterns) and danger signals from the dying cells cause an *immature* DC to become *mature*. Similarly, the signals from apoptotic cells transform an *immature* DC into a *semi-mature* DC. In both states, nevertheless, the DC is able to migrate and present the collected antigens in the *thymus* for *T-cell* activation.

Semi-Mature DCs. *Thymus* is the immune system organ where *T-cells* undergo maturation. The *semi-mature* DCs present their collected antigens to *T-cells* in *thymus* in a tolerogenic context. An exposure to safe signals during the antigen collection period causes the *semi-mature* DCs to secrete cytokines that leads to *T-cell* suppression. If receptors of the *T-cells* bind to the antigens presented by the *semi-mature* DCs, then they are de-activated, thus preventing the immune system to respond to these antigens.

Mature DCs. When an *immature* DC has had sufficient exposure to PAMPS and danger signals, it ceases to collect antigens and migrates to *thymus* as a *mature* DC. The cytokines secreted by *Mature* DCs have an immunogenic effect on the *T-cells* present in the *thymus*. If the *T-cell* receptors match any of the antigens presented by the *mature* DCs, the *T-cell* is activated. Activation of *T-cell* is needed to initiate the helper function for co-stimulation of B-cells during an adaptive immune response.

2.1 Applications of Danger Theory in AIS

The Danger Theory and dendritic cell behavior has found useful applications in the design and development of artificial immune systems, in general, and anomaly detection in particular. The authors in [1] and [2] discuss how the latest immunological concepts proposed by the danger theory may be mapped to solve the intrusion detection problem in computer security. They focus on identifying various (danger) signals and carry out their functional analysis to drive the adaptive immune response. The Danger Project [14] resulted in the development of the Dendritic Cell Algorithm (DCA), which was introduced in [6] as an abstract model for the dendritic cells interactions and behavior. The algorithm features all the major behavioral aspects of the dendritic cells: (1) the ability to sample multiple antigens, (2) process signals, (3) express co-stimulatory molecules and output cytokines, (4) adopt differentiation pathways, and (5) present the antigens in an appropriate context. The preliminary results indicate the suitability of the algorithm for anomaly detection. The authors conducted additional experiments [7] on a machine learning dataset and detection of outgoing portscans. The conclusion of the work is that the DCA has a potential to act as a classifier for static machine learning dataset, therefore, it can act as an anomaly detector in real network environments.

3. BEEAIS: ARTIFICIAL IMMUNE SYSTEM SECURITY

BeeAIS [11] is our first AIS based security framework for the *BeeAdHoc* protocol. It is based on self/non-self discrimination and performs anomaly detection using the *negative selection*. *BeeAIS* first learns the normal behavior of the system during an initial *learning* phase of 50 seconds, and then monitors the system for occurrences of abnormal patterns. The system, therefore, has the ability to detect previously unknown attacks.

Static Node Simulations. In [11], we compared the security characteristics of *BeeAIS* with its base protocol *BeeAdHoc* and the cryptographic security framework, *BeeSec* [12]. The simulations were performed in ns-2 on a static grid of 49 nodes. A static scenario was selected to make it easier to show the effect of attacks; with node mobility it becomes harder to demonstrate the attacks effects. We demonstrated that *BeeAIS* was able to detect a number of routing attacks.

Using the same simulation scenario, we further evaluated the impact of security enhancements on the performance of *BeeAIS*. The ns-2 simulation results showed that the performance metrics for *BeeAIS* were close to that of *BeeAdHoc* and better than that for *BeeSec*. It was, therefore, concluded that *"the AIS based security does not appreciably degrade the system performance compared to the original algorithm even though it provides the same level of security as of cryptographic approach"*.

3.1 BeeAIS simulations under mobility

In this section, we evaluate the performance of the *BeeAIS* protocol under mobility and show that the *BeeAIS* self/non-self model cannot adapt to a changing self due to mobility in MANETs. We perform simulations in ns-2 to compare the average throughput of *BeeAIS* with that of its base protocol *BeeAdHoc* and also with the classical MANET routing protocols, *DSR* and *AODV*. We deploy 10 to 60 nodes in a rectangular area of operation, 2400 × 480 m^2. Each experiment lasts 1000 seconds. Node movement is according to the "random waypoint" model. Each node in the network sends and receives data, comprising constant bit rate (CBR) peer-to-peer traffic at the rate of 30 packets/second. The results are averaged over five independent runs to factor out the stochastic elements.

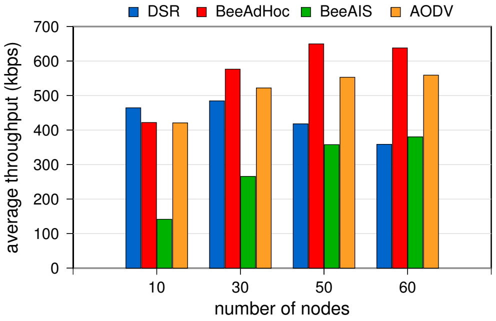


Figure 1: Comparison of protocol average throughputs

We define the protocol average throughput as, *"the total number of data bits delivered to destination nodes during the simulation, divided by the total simulation time"*. We computed the average throughput of *BeeAIS*, *BeeAdHoc*, *DSR* and *AODV* protocols. Figure-1 shows that the *BeeAIS* has the smallest average throughput. We investigated the reason for such a poor performance by logging the average number of data bytes handed down to the different protocols by the transport layer, along with the average number of data packets dropped by the protocols during the course of the simulation. Figure-2 shows that the data bytes received by the protocols from the transport layer that is normalized with respect to the highest value. We see that the *BeeAIS* receives the minimum amount of data amongst all the protocols (39.8% to 65.6% less than the *BeeAdHoc* protocol). Moreover, it dropped a higher number of data packets (see Table-1). Compared to the *BeeAdHoc* protocol, *BeeAIS* dropped from 28.1% to 145.5% more packets. Consequently, *BeeAIS* has the lowest average throughput among all the compared protocols.

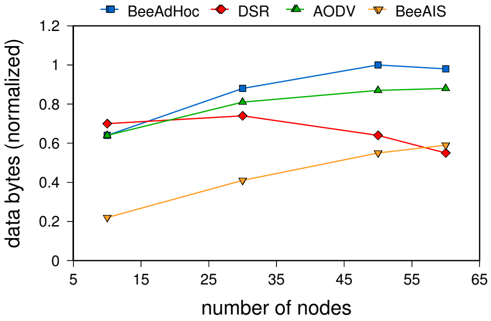


Figure 2: Comparison of application data handed down by TCP layer for transporting to destination nodes

Investigating the high data packet drop by *BeeAIS*, we measured the *BeeAIS* ability to detect the *self antigens (Ags)* as *self* when the frequent node movements cause the system self to change. Our results are shown in Table-2 for four different network scenarios. For each scenario, we measured the average number of antigens (scout Ags, Type-I forager Ags and Type-II forager Ags). For definitions refer to [11]. Since these simulations do not involve generating routing attacks, all the antigens are self Ags. We determine the average number of self Ags detected as non-self Ags (false positive (FP)) and as self Ags (true negative (TN)). We then compute the false alarm rate (FAR), i.e the percentage of self Ags that are detected as non-self Ags. Our results in Table-2 indicate that the scout Ags has a high FAR, resulting in as many as 67.36% scouts to be dropped in small MANETs; the figure drops to 12.84% with an increase in the node density and higher node connectivity.

When scouts are dropped, new routes are not discovered and the nodes drop the foragers due to *"route not available"*. The dropping of foragers fools the Transmission Control Protocol (TCP) to initiate congestion control when its retransmission timer expires. The sending TCP thus reduces its congestion window and enters the slow start phase. This causes a reduction in the amount of data handed down by the TCP layer to the network layer for routing that in fact decreases the average throughput of the network.

3.2 BeeAIS Mobility Limitation

The lowest average throughput of the *BeeAIS* protocol under mobility can be attributed to the *BeeAIS* initial *learning* phase of 50 seconds. The detector sets generated are based upon the normal behavior (system self) learned during this phase. Therefore, it can not adapt its detectors set to the changing self or non-self due to

Table 1: Comparison of data packets generated by applications and dropped by protocols

Protocol	*Average number of packets*	*Number of nodes*			
		10	30	50	60
Beeadhoc	*generated by application*	100864.20	138241.80	156178.00	153409.00
	dropped - route not available	228.20	469.20	632.80	595.80
	dropped per 1000 generated	**2.26**	**3.39**	**4.05**	**3.88**
AODV	*generated by application*	100722.00	126313.60	134829.00	136482.20
	dropped - route not available	189.60	186.00	123.60	88.00
	dropped per 1000 generated	**1.88**	**1.47**	**0.92**	**0.64**
Beeais	*generated by application*	33930.60	63935.00	86204.40	91599.60
	dropped - route not available	188.40	405.60	477.60	454.80
	dropped per 1000 generated	**5.55**	**6.34**	**5.54**	**4.97**

Table 2: BeeAIS: Detection of self Ags as non-self Ags due to mobility

Number of nodes	*Ag type*	*Avg Ags rcvd*	*Avg Ags Detected*		FAR
			FP	TN	(% age)
10 nodes	***scout Ags***	586.40	395.00	191.40	**67.360**
	forager Ags Type-I	30654.80	4.80	30650.00	0.015
	forager Ags Type-II	38312.50	398.25	37914.25	1.039
30 nodes	***scout Ags***	8297.00	2542.20	5754.80	**30.639**
	forager Ags Type-I	58911.20	38.00	58873.20	0.064
	forager Ags Type-II	58873.20	1099.80	57773.40	1.868
50 nodes	***scout Ags***	14106.00	2247.60	11858.40	**15.933**
	forager Ags Type-I	81798.80	51.20	81747.60	0.062
	forager Ags Type-II	81747.60	3034.00	78713.60	3.711
60 nodes	***scout Ags***	16804.60	2157.40	14647.20	**12.838**
	forager Ags Type-I	86718.00	51.00	86667.00	0.058
	forager Ags Type-II	86667.00	6873.00	79794.00	7.930

mobility. Consequently, the new and changed self is classified as non-self, and the relevant antigens, scouts or foragers, are dropped. Therefore, in order to allow mobility in *BeeAIS*, the system needs to incorporate a dynamic detectors set, which keeps evolving with the changing *self*, to allow for changes in the system *self* and the *non-self* space.

4. BEEAIS-DC: A DENDRITIC CELLS INSPIRED AIS SECURITY FRAMEWORK

BeeAIS-DC is a danger theory inspired AIS security framework, which is our third approach towards securing the *BeeAdHoc* protocol, after the *BeeSec* [12] and *BeeAIS* [11]. The proposed framework models the dendritic cells that provide it with the capability of learning the system *self* and the *non-self* that keeps changing with the node mobility in a MANET environment. Therefore, *BeeAIS-DC* overcomes a serious shortcoming of *BeeAIS*, as discussed in Section 3.2.

BeeAIS-DC utilizes a dynamic detector set that is mediated through the dendritic cells. We model the dendritic cells to sample the Ags (scouts) from the body tissues (the *node*). The sampling includes both the *self* and the *non-self* Ags. Later on depending on the presence or absence of *danger signal*, the dendritic cells follow the differentiating pathways towards their terminal states, *mature* or *semi-mature*, before presenting the sampled Ags for *T-cell* (detector) maturity in the *thymus*. The use of the *danger signal* in *BeeAIS-DC* precludes the need for an initial *learning phase* at system start up. Moreover, the absence of *danger signal* allows the changed normal behavior of the system to be presented as a new self, instead of being interpreted as non-self. The key feature of the system, therefore, is its ability to differentiate *self* from *non-self* quite early in the system's operations, and also to adapt to a changing self and non-self environment. Details of the *BeeAIS-DC* algorithm are explained in the coming sections.

4.1 Antigens

In *BeeAIS-DC*, we adopt the same Ag format as in [11]. An Ag is formed whenever a node receives a *forward* scout or a *backward* scout. The relevant header fields are extracted from the *scouts* that comprise the quadruple $\langle S_{sct}, D_{sct}, RtLen, node_{i-1} \rangle$. Antigens are represented in binary hamming shape space and have a string length of 52 bits each. An Ag has four genes, having lengths *16, 16, 4 and 16* bits and each gene represents a header field value. All the four collected genes are then concatenated to form an Ag.

4.2 Dendritic Cell (DC) Formation

When a scout is seen by a node for the first time, a dendritic cell is instantiated. At its first incarnation, several attributes of the dendritic cells need to be initialized that support their functionality later on. These include:

DC Ag. The scout Ag is added to the dendritic cell. This represents the Ag sampled from the tissue that is later presented to the *T-cells* during their maturation in the *thymus*.

DC Life. The dendritic cells live in the system for a short duration and then die. This ensures that the most recent system state is always presented in *thymus*. Moreover, it facilitates correct interpretation of the current system *self* and *non-self*.

DC State. The state of a DC may be *immature*, *semi-mature* or *mature*. At first incarnation, a dendritic cell is *immature*. When it samples the Ag and is exposed to *safe signals* it makes a transition to the *semi-mature* state. On the other hand if it is exposed to *danger signals*, then it changes to *mature* state. It can then migrate to the *thymus* to present the sampled Ag.

On receiving a scout, the node needs to determine whether the same bee agent was processed earlier or not. The node, therefore, matches the S_{sct}, D_{sct} and the complete source route with the respective values in the collected scout dendritic cells.

Table 3: List of BeeAIS-DC symbols and parameters

Symbol/Parameter	Description
S_{sct}, D_{sct}, $RtLen$, $node_{i-1}$	scout source, scout destination, source route length and the previous node address
Ag, T_{curr}, DC_{sct}	antigen, current time during simulation and scout dendritic cells
$Count_{FS}$, $Count_{BS}$	number of forward and backward scouts recvd
UDINT	fixed small interval of time defined for the system such that after each UDINT period the system checks for occurrences of danger signals and updates the dynamic detector set
THRESH-RCVD-FS, THRESH-RCVD-BS	upper limit for average forward or backward scouts to be received by a node before the context can be declared as dangerous
CO-STIMUL-SCT	co-stimulatory threshold for transition of dendritic cell state to MATURE, to allow presentation of the sampled non-self Ag in thymus for detector generation
NUM-DETS-SCT	number of detectors (antibodies) maintained by the system at any given time for matching the incoming scout Ags

The life of a newly instantiated scout dendritic cell is determined by Equation-1. In case a similar scout reappears (matches an existing scout DC), the scout DC life is reset using the same equation. Also the count for receiving a *forward* or a *backward* scout ($Count_{FS}$ or $Count_{BS}$) for the matching DC is incremented. This information is used later by *BeeAIS-DC* to determine the occurrence of scout *danger signal* as explained in Section 4.3.

$$DC_{sct}\, life \;=\; T_{curr} \;+\; UDINT \qquad (1)$$

4.3 Danger Signal Computation

The most important step of dynamically updating the detector's set is computation of *danger signal* by a node. A *danger signal* in BIS occurs when there is an evidence of necrotic cell death in the *tissue*, indicating damage to body due to pathogenic infection. In a mobile network scenario, damage to a network may be an irregular and inefficient routing behavior. Therefore, if there is an evidence of routing problems in the network, the relevant *danger signal* might be raised.

DCs need activation by a *danger signal* to change their state to *mature* before migration to *thymus* and presentation of the sampled Ags as *non-self.* The absence of *danger signal* results in a *semi-mature* state for the DCs with the sampled Ags regarded as self. Identification of a suitable *danger signal* in the network should thus allow incorporation of the changed and most recent *self* and *non-self* states in the system in order to continuously update the detectors set.

In *BeeAIS-DC*, computation of *danger signal* and accordingly updating the detectors set is done at fixed and periodic intervals that are *update interval* (UDINT) seconds apart. During regular operation, the scout DCs at each node keep count of the forward and backward scouts received by the node during the last UDINT period of time. At the end of every UDINT period, the DCs determine the average forward and backward scouts received on each of the stored paths. Each time the computed averages exceed their respective thresholds (THRESH-RCVD-FS, THRESH-RCVD-BS), the co-stimulation level for that path is raised. Finally, when the co-stimulation level exceeds the co-stimulation threshold for scouts (CO-STIMUL-SCT), the *danger signal* for that scout is turned "HIGH". Now, the context for this DC becomes "dangerous", the DC turns its state to *mature* and migrates to *thymus* to present its sampled Ag as a *non-self* scout Ag.

Equation-1 indicates that a scout DC can survive a minimum of two consecutive UDINT time periods before undergoing a natural death. Within this period, however, if a similar scout arrives again, the DC life is increased for another UDINT period. This implies that the *danger signal* would turn high only if the *non-self* Ag keeps arriving within the life span of a DC, it is detected as suspected and the number of UDINT intervals within which the Ag is detected as suspected exceeds the threshold. This provides sufficient co-stimulation before raising the *danger signal* and helps to reduce the rate of false positives.

Once raised, the *danger signal* remains high for the subsequent (CO-STIMUL-SCT + 1) number of UDINT periods. This reduces the false negatives if the attack continues but skips detection in contiguous UDINT periods.

4.4 Detector Set Updation

In *thymus*, the process of *T-cell* maturation takes place in the presence of DCs. If the sampled Ag is presented by the DC in *semi-mature* state, the *T-cells* that match the Ag die. In other words, a population of *T-cells* is generated that is tolerant to *self* through elimination of *T-cells* whose antibodies match the self Ags. If the sampled Ag is presented in the *mature* state, the matching *T-cells* get activated and are then ready to help initiate the adaptive immune response.

On the same principle, in *BeeAIS-DC*, the purpose of the Ag sampling by DCs and determination of their states as *semi-mature* or *mature* is to affect the creation of antibodies or detector set for scouts that are *self* tolerant. The resulting detectors would then recognize or match only the *non-self* Ags and the activated ones would assist in initiating the required response against non-self Ags.

At system startup, random detectors are generated to produce the required scout detectors set. It is also subjected to negative selection with respect to the *self* Ags presented by the DCs in *semi-mature* state. Sometimes the existing detectors set is used. If any detector matches a *self* Ag, it is removed from the system. This may result in the number of scout detectors to fall below the level (NUM-DETS-SCTS) specified for the system. Therefore, to make up for the lost detectors, the node again generates random detectors and adds only those to the scout detectors set that do not match the *self* Ags. This ensures that the resulting detectors are capable only of binding with the *non-self* Ags.

The Ags, sampled by DCs, which have "dangerous" context, are presented by the DCs in *mature* state. The *mature* DCs activate those *T-cells* that match the sampled *non-self* Ag. We model this activation of *T-cells* by matching all detectors with the Ags presented by DCs in *mature* state and changing the state of matching detectors from *naive* to *activated*. Now if any of the incoming Ags match these *activated* detectors, the Ags are detected as *non-self.*

4.5 Eliminating or Refreshing DCs

We need to update the scout DCs after every UDINT period of time, when the scout detectors set has been updated. The DCs that die a natural death, i.e completed their lives during the last UDINT period, need to be eliminated from the system. The surviving DCs are then refreshed to restart the process of Ag sampling in tissues and determining the occurrence of *danger signal*. The state of the surviving *semi-mature* and *mature* dendritic cells is changed to *im-*

mature. Moreover, the data gathering fields of the DCs ($Count_{FS}$, $Count_{BS}$) are also reset for collection of new data in the following UDINT period.

4.6 Matching Antigens and Detectors

A node, during normal operations, receives the bee agents and then classifies them as *self* or *non-self* after matching the scout Ags with the scout detectors set. If a scout Ag matches with an activated scout detector, a *non-self* Ag is assumed to have been identified and the matching scout is dropped.

5. DEMONSTRATING ATTACK EFFECTS

We implemented *BeeAIS-DC* in network simulator, ns-2, and then validated its security functionality through simulations. We used the same simulation scenario as in [12] that consisted of nine nodes in a simple topology, and performed the attack simulations on *BeeAIS-DC* to demonstrate the behavior of the protocol under the following conditions:

- **Normal Routing Behavior.** Fully functional *BeeAIS-DC* protocol working in normal conditions without any attack nodes.
- **Partially Functional Under Attack.** *BeeAIS-DC* with secure scout processing but not dropping them if detected as a *non-self* Ag. Consequently, malicious nodes will successfully launch attacks.
- **Fully Functional Under Attack.** Fully functional *BeeAIS-DC* that drops scouts once detected as *non-self* Ags.

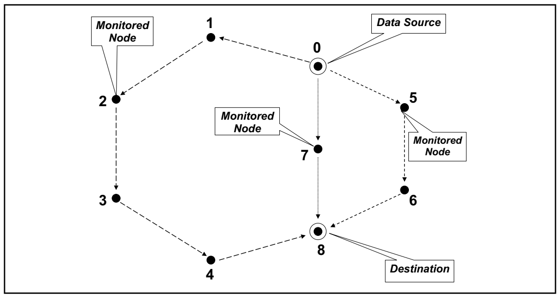


Figure 3: Node topology selected for attacks

5.1 Node Topology for Attacks

The network topology selected for demonstrating the effect of attacks is shown in Figure-**??**. It is a rectangular area of 1000 × 500 m^2, where *Node-0* is the source and *Node-8* is the destination. The source *Node-0* has a TCP traffic source that generates constant bit rate (CBR) data traffic for the sink connected to the destination *Node-8*.

In Figure-**??**, we can see three distinct paths between the source and the destination nodes: *0-7-8*, *0-5-6-8* and *0-1-2-3-4-8*. The path *0-1-2-3-4-8* being the least optimum should virtually have no packets routed over it under normal conditions.

5.2 Attacks on BeeAIS-DC

We developed an *attacker framework* in ns-2 in order to launch routing attacks on *BeeAIS-DC* protocol. It is capable of launching two different types of routing attacks. During attacks we monitored the routed traffic at three points in the network: *Node-2*, *Node-5* and *Node-7*. We then generated traffic maps to indicate the success or failure of these attacks.

Attack-1: Forging Forward Scout. This attack was launched at time t=100 seconds after the start of the simulation. The attacker *Node-4* launched forged forward scouts into the network and tried to install a forged route *0-1-2-3-4-8*. These fake packets have *Node-0* as the S_{sct} and *Node-8* as the D_{sct}. Figure-4(b) shows the results of the attack when *BeeAIS-DC* is running with partial functionality. In this case, as the forward scouts are received at D_{sct}, *Node-8*, they are returned to the S_{sct} *Node-0*, resulting in the forged route to be established. Subsequently, all foragers started to follow the forged route and the attack is successful.

However, in the case of *BeeAIS-DC* running with full functionality, the attack is not successful (see Figure-4(c)). At D_{sct} *Node-8*, when the arrival rate of forged forward scouts exceeded the threshold in more than the CO-STIMUL-SCT number of contiguous update intervals, the *danger signal* due to forward scouts was turned high. The relevant DCs then made a transition to *mature* state and presented the sampled forged scout Ags as *non-self* Ags for updating the scout detectors set. Finally, the scout detectors were able to match the forged scouts and dropped them to make the attack unsuccessful. Therefore, the *BeeAIS-DC* successfully countered the attack of the malicious node (see Figure-4(a)).

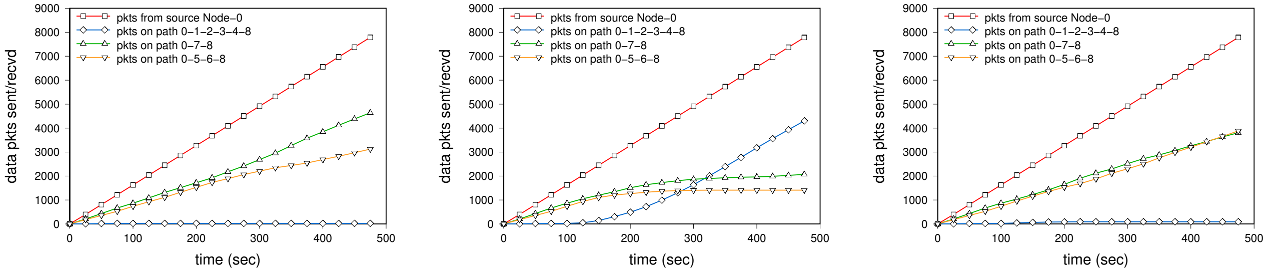
Attack-2: Forging Backward Scout. The attack involved spoofed backward scouts and was launched by *Node 2* at time t=100 seconds. The attack was successful in the case of *BeeAIS-DC* running with partial functionality due to the forged path *0-1-2-3-4-8* getting established at S_{sct} *Node 0*. As shown in Figure 5(b), the malicious *Node 2* was able to divert the data packets on the forged path. But when attack was launched once *BeeAIS-DC* was running with full functionality, the D_{sct} *Node 8* was able to detect the forged backward scouts as *non-self* Ags and dropped them. Consequently, as seen in Figure 5(c), the forged path *0-1-2-3-4-8* was not established and the routing behavior of the protocol remained the same as in the case of *BeeAIS-DC* without attacks (see Figure-5(a)).

6. NETWORK PERFORMANCE

We compared the network performance of our proposed security framework, *BeeAIS-DC*, through extensive simulations in the network simulator ns-2 with the *BeeAdHoc* protocol. We used the same simulation scenario as in Section-3.1. The performance metrics used are:

Throughput. *The number of data bits delivered to the application layer at the destination node in a unit interval of time.*
Packet Delivery Ratio. *The ratio of data packets successfully delivered to destination nodes and total number of packets generated for those destinations.*
Latency. *The average difference in time when a packet is generated at the source node and when it got delivered to the destination node.*
Average Hops. *The average number of hops for all the paths traversed by data packets.*
Transmission efficiency. *The number of data bytes delivered to the application layer at destination nodes at the cost of a unit control byte.*
Control Byte Overhead. *Total number of control bytes transmitted by all nodes in the network.*

Figure-6 shows the results for our extensive performance evaluation of the *BeeAIS-DC* algorithm. We have compared the performance of the security framework with its base protocol *BeeAdHoc* as well as with the *AODV* and *DSR*. We can see that the AIS overhead of

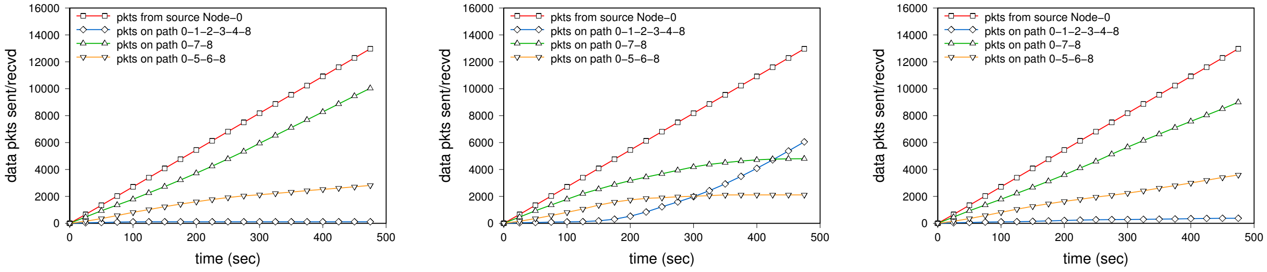


(a) Normal routing without attacks

(b) Under attack without AIS pkt dropping

(c) Under attack with AIS pkt dropping

Figure 4: BeeAIS-DC Attack-1: Forging Forward Scout



(a) Normal routing without attacks

(b) Under attack without AIS pkt dropping

(c) Under attack with AIS pkt dropping

Figure 5: BeeAIS-DC Attack-2: Forging Backward Scout

BeeAIS-DC does not significantly degrade the performance of the *BeeAdHoc* protocol; all performance parameters for both *BeeAdHoc* and *BeeAIS-DC* are nearly the same. At the same time, we can see that in most of the cases, *BeeAIS-DC* outperforms even the non-secure state-of-the-art classical MANET routing protocols, *AODV* and *DSR*. Therefore, our proposed security framework is able to provide protection against the routing attacks but it delivers the same or slightly better network performance as compared to the other MANET routing protocols.

We also compared the network average throughput of all the protocols (see Table-4). The average throughput for *BeeAIS-DC* is quite close to that of *BeeAdHoc*. This indicates that the *BeeAIS-DC* did not suffer from the changing self problem due to mobility the way *BeeAIS* did. The DCs were correctly able to classify the newly changed system *self* and it did not drop self scouts.

Protocol	*Number of Nodes*			
	10	30	50	50
BeeAdHoc	421.86	576.36	649.67	637.91
DSR	464.50	484.54	418.01	358.72
AODV	420.81	522.00	553.02	559.09
BeeAIS-DC	419.68	570.85	656.62	660.09

Table 4: Average network throughputs for protocols (kbps)

7. CONCLUSION AND FUTURE WORK

In this paper we have introduced a secure routing framework, *BeeAIS-DC*, for the Bio/Nature inspired MANET routing protocol, *BeeAdHoc*. Our proposed framework is based upon the Danger Theory, which models the function and behavior of the dendritic cells in the Biological Immune System. We first showed that the self/non-self based system, *BeeAIS*, suffers from a poor average network throughput. This is because the frequent node movements in a MANET environment results in the system *self* to change and the non-adaptive detectors set of *BeeAIS* still detects the newly changed *self* as *non-self*. To overcome this limitation, we implemented our danger signal based AIS framework, *BeeAIS-DC*. Our proposed system utilizes the principles of the dendritic cells to provide the capability of dynamically updating the detectors set to cater for a changing system *self* and *non-self*. By sensing the presence/absence of danger, *BeeAIS-DC* is able to differentiate between the newly changed *self* and the malicious *non-self* behavior.

We extensively evaluated the network performance of *BeeAIS-DC* in ns-2 simulator. We compared its performance with *BeeAdHoc*, *DSR* and *AODV* protocols. Our results show that the *BeeAIS-DC* provides protection to mobile nodes in a MANET against routing attacks of malicious nodes. Its communication and computation overhead is minimal that ensures that its performance is comparable to the *BeeAdHoc* protocol. In future, we want to extend our *BeeAIS-DC* to cater for attacks related to tampering/forging of foragers and also to detect the dropping attacks in MANETs.

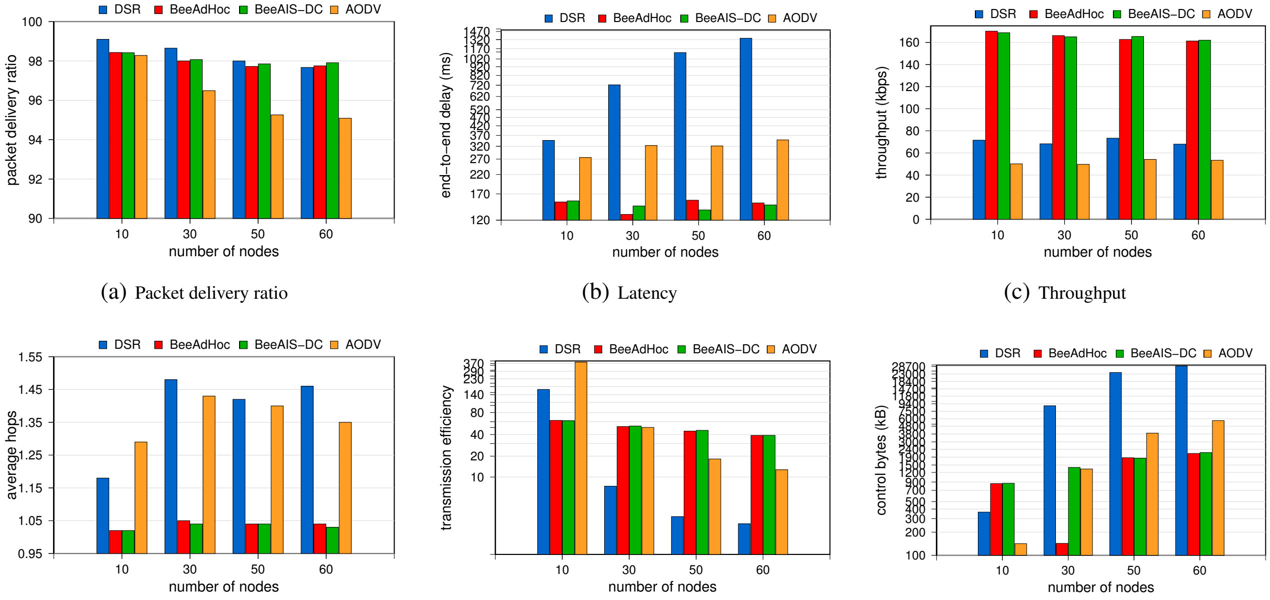
8. REFERENCES

[1] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod. Danger theory: The link between ais and ids. In *Proceedings of the ICARIS-2003, LNCS 2728*, pages 147–155, 2003.

[2] U. Aickelin and S. Cayzer. The danger theory and its applications to artificial immune systems. In *Proceedings of the ICARIS-2002*, pages 141–148, 2002.

[3] U. Aickelin, J. Greensmith, and J. Twycross. Immune system approaches to intrusion detection - a review. In *Proceedings of the ICARIS-2004, LNCS 3239*, pages 316–329, 2004.

[4] G. Di Caro, F. Ducatelle, and L.M. Gambardella. Anthocnet: An adaptive nature inspired algorithm for routing in mobile



(a) Packet delivery ratio

(b) Latency

(c) Throughput

(d) Average hops

(e) Transmission efficiency

(f) Control byte overhead

Figure 6: Performance results comparing BeeAIS-DC with BeeAdHoc, DSR and AODV

ad hoc networks. *European Transactions on Telecommunications*, 16(2):443–455, 2005.

[5] L. N. de Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer, 2002.

[6] J. Greensmith, U. Aickelin, and S. Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *Proceedings of the ICARIS-2005, LNCS 3627*, pages 153–167, 2005.

[7] J. Greensmith, J. Twycross, and U. Aickelin. Dendritic cells for anomaly detection. In *Proceedings of the CEC*, pages 664–671, 2006.

[8] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11(1-2):21–38, 2005.

[9] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, pages 153–181. 1996.

[10] P. Matzinger. Tolerance, danger, and the extended family. *Annual Review of Immunology*, 12:991–1045, 1994.

[11] N. Mazhar and M. Farooq. Beeais: Artificial immune system security for nature inspired, manet routing protocol, beeadhoc. In *Proceedings of ICARIS-2007, LNCS 4628*, pages 370–381, Aug, 2007.

[12] N. Mazhar and M. Farooq. Vulnerability analysis and security framework (beesec) for nature inspired manet routing protocols. In *Proceedings of GECCO-2007*, pages 102–109, July, 2007.

[13] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[14] Danger Project. http://www.dangertheory.com.

[15] M. Roth and S.Wicker. Termite: Ad-hoc networking with stigmergy. In *Proceedings of IEEE GLOBE-COM*, Dec 2003.

[16] S. Sarafijanovic and J.Y. Le Boudec. An artificial immune system approach with secondary response for misbehavior detection in mobile ad-hoc networks. *IEEE Transactions on Neural Networks*, 16(5), Sep 2005.

[17] H.F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat. Beeadhoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *GECCO*, pages 153–160, 2005.

[18] H.F. Wedde, C. Timm, and M. Farooq. Beehiveais: A simple, efficient, scalable and secure routing framework inspired by artificial immune systems. In *PPSN*, pages 623–632, 2006.

[19] Manel Guerrero Zapata. Secure ad hoc on-demand distance vector (saodv) routing. Internet-Draft, draft-guerrero-manet-saodv-05.txt, February, 2005.