An Error Propagation Algorithm for Ad Hoc Wireless Networks

Martin Drozda, Sven Schaust, Sebastian Schildt and Helena Szczerbicka

Simulation and Modeling Group
Dept. of Computer Science
Leibniz University of Hannover
Welfengarten 1, 30167 Hannover, Germany
Email: {drozda, svs, hsz}@sim.uni-hannover.de, sebastian.schildt@reparts.org

Abstract. We were inspired by the role of co-stimulation in the Biological immune system (BIS). We propose and evaluate an algorithm for energy efficient misbehavior detection in ad hoc wireless networks. Besides co-stimulation, this algorithm also takes inspiration from the capability of the two vital parts of the BIS, the innate and the adaptive immune system, to react in a coordinated way in the presence of a pathogen. We demonstrate that this algorithm is also applicable in situations when a (labeled) data set for learning the normal behavior and misbehavior is unavailable.

1 Introduction and Motivation

Ad hoc and sensor wireless networks can become an object of attacks and intrusions. The motivation for attacking an ad hoc network can range from a desire to benefit from the network's services to an intent to make it non-functional. Faults that are a result of software or hardware failures can be equally damaging. Correcting the consequences of some faults or attacks might only be possible by a costly human intervention, or not at all. Even though secure protocols address issues connected with data integrity and user authentication, the experience with the Internet shows that flaws in these protocols are continuously being found and exploited [1].

This establishes the basic motivation for designing autonomous detection and response systems that aim at offering an additional line of defense to the employed secure protocols. Such systems should provide several layers of functionality including the following: (i) distributed self-learning and self-tuning with the aspiration to minimize the need for human intervention and maintenance, (ii) active response with focus on attenuation and possibly elimination of negative effects of faults or attacks on the network.

In many scenarios ad hoc and sensor networks are expected to be based on wireless devices with limited (battery) resources. In order to stimulate the survivability of such networks, it is essential that autonomous detection and response systems reflect these resource constraints.

Best current practices for misbehavior detection in ad hoc wireless networks are almost exclusively done on a domain knowledge basis; see [2] and references therein. Although such an approach allows to find a good predictor for a specific type of misbehavior, it fails to address a broader range of issues. Furthermore, the area of energy efficient misbehavior detection remains to be a challenging problem.

We assume that upon deployment of an ad hoc network, an enforcement of operational strategies in an energy efficient way is desired. Such operational strategies may impose performance limits in the form of e.g. a maximum data packet loss at a node. One possibility for determining whether a node keeps forwarding data packets is monitoring by other nodes. This is however a very costly approach since the monitoring nodes cannot enter a sleep mode to preserve their energy resources. It is therefore necessary that monitoring will be kept to a minimum.

Our aim was to design a system offering a reasonable tradeoff between energy efficiency and misbehavior classification performance. Our contribution is an algorithm that allows for energy efficient misbehavior detection that can also be applied to e.g. just deployed ad hoc networks. Such ad hoc networks have none or a very limited data basis for an efficient learning of the normal behavior and misbehavior. To stimulate energy efficiency, we exploit an inherent property of ad hoc networks in order to propagate the classification error among two parts of our detection system. These two parts are inspired by the role of the innate and adaptive immune systems and by their ability to communicate. Such error propagation allows for an adaptive approach to the costly explicit behavior monitoring by the participating nodes.

2 The Biological Immune System

The Biological immune system (BIS) [3] can quickly recognize the presence of foreign microorganisms in the human body. It is remarkably efficient, most of the time, in correctly detecting and eliminating pathogens such as viruses, bacteria, fungi or parasites. When confronted with a pathogen, the BIS often relies on a coordinated response from both of its two vital parts:

- the *innate system*: the innate immune system is able to recognize the presence of a pathogen or tissue injury, and is able to signal this to the adaptive immune system.
- the *adaptive system*: the adaptive immune system can develop during the lifetime of its host a specific set of immune responses and provide immunological memory. An immunological memory serves as a basis for a stronger immune response, should a pathogen re-exposure happen.

The form and amplitude of immune responses is pathogen dependent. In many forms of immune reactions, the innate immune system plays an important role in triggering a reaction from the adaptive immune system, and vice versa. A cell gets eliminated, if it was classified by the adaptive immune system as

a pathogen, and at the same time, the innate immune system signals that the cell causes some damage to the human organism. Under this specific immune reaction, only damage inflicting or *infectious* cells get eliminated by the BIS. This implies that a two-way communication, hereafter referred to as *co-stimulation*, between the innate and adaptive immune systems is necessary.

Communication capabilities within the BIS received an increased interest from the Artificial immune systems (AIS) community and evolved into an independent research direction. Several different types of danger, safe and amplifying signals were proposed within the Danger theory due to Aickelin et al. [4].

In our work, we were motivated by the ability of the BIS to act in a coordinated way when confronted with a pathogen. We show that co-stimulation in ad hoc networks can have very positive effects in stimulating energy efficiency.

3 Related Work

The pioneering work in adapting the BIS to networking has been done by Stephanie Forrest and her group at the University of New Mexico. In one of the first BIS inspired works, Hofmeyr and Forrest [5] described an AIS able to detect anomalies in a wired TCP/IP network. Co-stimulation was in their setup done by a human operator who was given 24 hours to confirm a detected attack.

Sarafijanović and Le Boudec [6] introduced an AIS for misbehavior detection in mobile ad hoc wireless networks. They used four different features based on the network layer of the OSI protocol stack. They were able to achieve a detection rate of about 55%; they only considered simple packet dropping with different rates as misbehavior. A co-stimulation in the form of a danger signal emitted by a connection source was used to inform nodes on the forwarding path about perceived data packet loss. Such signal could then be correlated with local detection results and suppress false positives.

An AIS for sensor networks was proposed by Drozda et al. in [7]. The implemented misbehavior was again simple packet dropping; the detection rate was about 70%.

Classification techniques proposed in [5–7] are based on negative selection, a learning mechanism applied in training and priming of T-cells in the thymus. In the computational approach to negative selection due to D'haeseleer et al. [8], a complement to an n-dimensional vector set is constructed. This is done by producing random vectors and testing them against vectors in the original vector set. If a random vector does not match anything, according to some matching rule in the original set, it becomes a member of the complement (detector) set. These vectors are then used to identify anomalies (faults/misbehavior). Timmis et al. [9] recently showed that negative selection is NP-complete, if the n-dimensional vectors are represented as bit-vectors and the matching/testing is done using the r-contiguous bits matching rule [8]. In general, the negative selection process, in the current interpretation, does not seem to have the potential to be used for misbehavior detection in ad hoc wireless networks.

An approach based on the Danger theory avoiding the inefficiency of the negative selection was proposed by Kim et al. in [10]. Several types of danger signals, each having a different function are employed in order to detect routing manipulation in sensor wireless networks. The authors did not undertake any performance analysis of their approach.

Drozda et al. [11] used a forward feature selection process together with a wrapper approach [12] to identify a suitable set of features for misbehavior detection. A co-stimulation inspired architecture with the aim to decrease the false positives rate while stimulating energy efficiency is proposed and evaluated.

Even though the BIS seems to be a good inspiration for improving misbehavior detection in ad hoc and sensor networks, approaches based on machine learning and similar methods received much more attention; see [2] and the references therein. Despite recent efforts, *energy efficient* misbehavior detection however remains to be a challenging problem.

4 Protocols and Assumptions

We now review several protocols, mechanisms, assumptions and definitions relevant to our experiments.

An *ad hoc network* can be defined as a net $N = (n(t), e(t))$ where $n(t), e(t)$ are the set of nodes and edges at time t, respectively. Nodes correspond to wireless devices that wish to communicate with each other. An edge between two nodes A and B is said to exist when A is within the radio transmission range of B and vice versa. A *sensor network* is a static ad hoc network deployed with the goal to monitor environmental or physical conditions such as humidity, temperature, motion or noise.

AODV is [13] is an on-demand routing protocol that starts a route search only when a route to a destination is needed. This is done by flooding the network with RREQ (= Route Request) control packets. The destination node or an intermediate node that knows a route to the destination will reply with a RREP (= Route Reply) control packet. This RREP follows the route back to the source node and updates routing tables at each node that it traverses. A RERR (= Route Error) packet is sent to the connection originator when a node finds out that the next node on the forwarding path is not replying.

At the MAC (Medium access control) layer, the medium reservation is often contention based. In order to transmit a data packet, the IEEE 802.11 MAC protocol uses carrier sensing with an RTS-CTS-DATA-ACK handshake (RTS = Ready to send, CTS = Clear to send, ACK = Acknowledgment).

In *promiscuous mode*, a node listens to the on-going traffic among other nodes in the neighborhood and collects information from the overheard packets. Promiscuous mode is energy inefficient because it prevents the wireless interface from entering sleep mode, forcing it into either idle or receive mode. There is also extra overhead caused by analyzing all overheard packets. According to [14], power consumption in idle and receive modes is about 12-20 higher than in sleep mode. Promiscuous mode requires that omnidirectional antennas are used.

We do not assume any node location knowledge or time synchronization among nodes. We assume that packets are authenticated, i.e. the sender of any packet can be easily identified as well as can be changes in the packet body. This is a reasonable assumption in line with e.g. the ZigBee specification [15].

5 Misbehavior Modeling and Classification Performance Evaluation

Node misbehavior can be the result of an intrusion or failure. We considered three types of misbehavior: (i) DATA packet dropping: 30% DATA packets were randomly and uniformly dropped at misbehaving nodes. (ii) DATA packet delaying: 30% DATA packets were randomly and uniformly delayed by 0.1 second at misbehaving nodes. (iii) Wormholes [16]. Wormholes are private (out-of-band) links between one or several pairs of nodes. They are added by an attacker in order to attract data traffic into them to gain control over packet routing. This could lead to packet data load manipulation.

In order to simplify the experiments, we decided to merge these three types of misbehavior into a single class called "misbehavior". This means, when evaluating the classification performance of our approach, we will apply a "normal vs. misbehavior" classification scheme. The necessary traffic samples for these two classes were created through network simulation by applying one of the above misbehavior models or running a misbehavior free simulation. Natural data packet losses (noise) due to e.g. collisions at the MAC layer amounted to $0 - 15\%$. The detailed experimental setup is discussed in one of the following sections.

Classification performance in our experiments was evaluated in terms of detection rate and false positives (FP) rate. The two measures were computed as follows:

$$det.\ rate = \frac{c_{c_j}}{n_{c_j}} \times 100.0\% \qquad FP\ rate = \frac{FP_{c_j}}{n_{c_j}} \times 100.0\% \tag{1}$$

where $c_j = \{normal, misbehavior\}$. n_{c_j} is the number of vectors (samples) labeled with the class c_j; note that $n_{c_j} > 0$ in all our experiments. c_{c_j} is the number of vectors that were correctly classified by the induction algorithm as belonging to the class c_j. FP_{c_j} is the number of samples incorrectly predicted as belonging to c_j. 95% confidence intervals ($CI_{95\%}$) were computed for each measure. Classification performance with respect to a vector set was evaluated by means of the classification error:

$$class.\ error = \frac{\sum_{c_j} FP_{c_j}}{\sum_{c_j} n_{c_j}} \times 100.0\% \tag{2}$$

6 Co-stimulation in Ad Hoc and Sensor Networks

Drozda et al. [11] introduced and evaluated an architecture inspired by the interplay between the innate and adaptive immune system. 24 features suitable

for misbehavior detection from several layers of the OSI protocol stack were considered. These features were divided into several subsets with respect to their energy requirements and protocol assumptions. A wrapper approach [12] was used to identify features with a *statistically significant contribution* to the classification process. Due to their relevance for our experimental setup, the features that were found significant are listed below.

6.1 The Features

Let $s_s, s_1, ..., s_i, s_{i+1}, s_{i+2}, ..., s_d$ be the path between s_s and s_d determined by a routing protocol, where s_s is the source node, s_d is the destination node. Features in the following feature set f are averaged over a time window. We use the feature labels (M3, M4, ...) as they were introduced in [11].

MAC Layer Features:

M3 **Forwarding index (watchdog):** Ratio of data packets sent from s_i to s_{i+1} and then subsequently forwarded to s_{i+2}.

M4 **Processing delay index (quantitative watchdog):** Time delay that a data packet accumulates at s_{i+1} before being forwarded to s_{i+2}.

Routing Layer Features:

R5 **Average distance to destination:** Average number of hops from s_i to any destination.

R9 **Connectivity index:** Number of destinations with known routes as recorded in the routing table of node s_i.

R12 **Diameter index:** Number of hops to the furthermost destination as recorded in the routing table of node s_i.

Transport Layer Features:

T1 **Out-of-order packet index:** Number of DATA packets that were received by s_i out of order. This assumes that each DATA packet has a unique ID computed by the connection source. Normalized by the time window size.

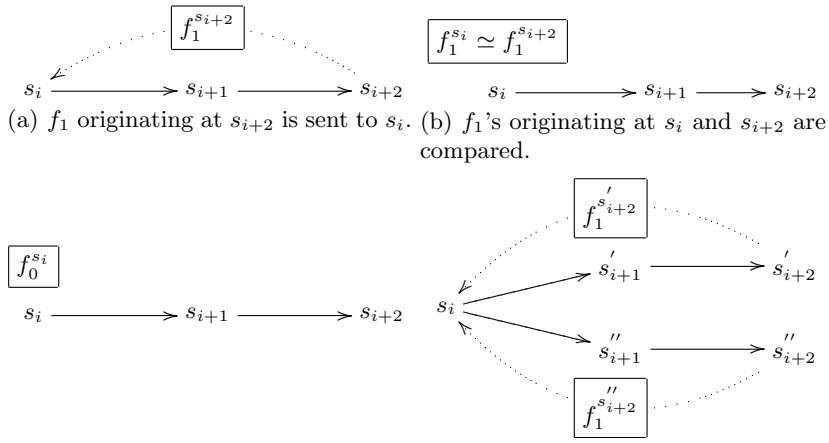
T2 **Interarrival packet delay index 1:** Average delay between DATA packets received by s_i. The delay was computed separately *for each connection* and then a master average was computed.

T3 **Interarrival packet delay variance index 1:** Variance of delay between DATA packets received by s_i. The variance was computed separately for *each connection* and then a master average was computed.

T4 **Interarrival packet delay index 2:** Average delay between DATA packets received by s_i.

T5 **Interarrival packet delay variance index 2:** Variance of delay between DATA packets received by s_i.

All the above features can be locally computed. The features T2, T3 and T4, T5 are identical, if only DATA packets belonging to a single connection are received by s_i. M3 and M4 require operation in promiscuous mode and therefore can be considered energy inefficient. Sudden changes in R5, R9 and R12 can help detect a topological change caused by a wormhole. This type of misbehavior



(a) f_1 originating at s_{i+2} is sent to s_i.

(b) f_1's originating at s_i and s_{i+2} are compared.

(c) *Co-stimulation:* if (b) indicates that s_{i+1} is misbehaving, f_0 will be computed and evaluated.

(d) If s_i forwards data packets for several connections, multiple f_1's are computed and sent.

Fig. 1. Immuno-inspired misbehavior detection approach.

increases the number of nodes lying within a fixed number of hops from s_i. We consider the following two subsets of the feature set f:

1. $f_0 = \{M3, M4\}$.
2. $f_1 = \{R5, R9, R12, T1, T2, T3, T4, T5\}$.

Besides energy efficiency, these two feature subsets have a very different expressive power. For example to show that s_{i+1} is misbehaving, either f_0 computed at s_i is necessary or f_1 computed at both s_i and s_{i+2} is necessary [11]. In other words, in case of f_1 measurements from several nodes are required, i.e. the output measured at s_{i+2} must be compared with the input at s_i in order to identify s_{i+1} as misbehaving. In the following, we use $f_0^{s_i}$, $f_1^{s_i}$ to denote f_0, f_1 computed by the node s_i, respectively.

6.2 A Co-stimulation Based Approach

To achieve a robust detection performance, a co-stimulation inspired mechanism is considered in [11]. This mechanism attempts to mimic the ability of communication between various players of the innate and adaptive immune system. More specifically, the inspiration was based upon the observation that a cell will not be marked as infectious and thus eliminated as long as no co-stimulation from other BIS players is received. Thus by analogy, a node should not mark another node as misbehaving as long as it does not receive a co-stimulatory signal from other parts of the network.

The co-stimulation inspired approach is depicted in Fig. 1. Each node in the network computes the feature set f_1. This feature set is then proliferated upstream (towards the connection source); see Fig. 1(a). Each f_1 travels exactly two hops; in our example from s_{i+2} to s_i. If s_i receives f_1 originating at one of its neighbors, it will forward f_1 in the upstream direction. If f_1 is not originating at one of its neighbors, it is used but not forwarded.

Since the computation of f_1 is time window based, the frequency with which f_1 gets sent depends on the time window size. Upon receiving f_1 from a two-hop neighbor, the node s_i combines and evaluates it with its own f_1 sample; see Fig. 1(b). Based on this, a behavior classification with respect to the node s_{i+1} is done. If s_i classifies s_{i+1} as misbehaving, then it computes a sample based on the f_0 feature set. This means, a *co-stimulation* from the f_1 based classification approach is needed in order to activate the energy less efficient f_0 based classification approach; see Fig. 1(c). If misbehavior gets confirmed, the node s_{i+1} is marked as misbehaving. Note that s_i can receive f_1 samples from several two-hop neighbors; see Fig. 1(d) for an example.

The proliferation of f_1 can be implemented without adding any extra communication complexity by attaching this information to CTS or ACK MAC packets. As long as there are DATA packets being forwarded on this connection, the feature set can be propagated. If there is no DATA traffic on the connection (and thus no CTS/ACK packets exchanged), the relative necessity to detect the possibly misbehaving node s_{i+1} decreases. Optionally, proliferation of f_1 can be implemented by increasing the radio radius at s_{i+2}, by broadcasting it with a lower time-to-live value or by using a standalone packet type.

If the node s_{i+1} decides not to cooperate in forwarding the feature set information, the node s_i will switch, after a time-out, to the feature set f_0 computation. In this respect, not receiving a packet with the feature set f_1 can be interpreted as a form of negative co-stimulation. If the goal is to detect a misbehaving node, it is important that the originator of f_1 can be unambiguously identified, i.e. strict authentication is necessary. An additional requirement is the use of sequence numbers for feature sets f_1. Otherwise, the misbehaving node s_{i+1} could interfere with the mechanism by forwarding outdated cached feature sets f_1.

We introduce the following notation in order to keep track of composite feature sets f_1 computed at the nodes s_i and s_{i+2}: $\mathcal{F}_1^{s_i} = f_1^{s_i} \cup f_1^{s_{i+2}}$. For simplicity, we will omit the superscript.

The mechanisms of the innate immune system bear a certain resemblance to the f_0 based classification phase; see Fig. 1(c). The innate system is for example very efficient in signaling tissue injury or damage to the adaptive immune system. To a certain degree it relies on some very rudimentary methods such as recognizing an unusually high level of dead or damaged self cells (e.g. blood cells). This can be directly compared with the very straightforward functionality of watchdogs. Similarly, the more learning extensive classification approach based on $\mathcal{F}_1$ (Fig. 1(b)) can be compared with the adaptive immune system.

6.3 An Error Propagation Algorithm for Ad Hoc Networks

In [11] it was concluded that as the size of the time window decreases, the classification ability of the feature sets f_0 and $\mathcal{F}_1$ will equalize. That is:

$$\lim_{win.\ size \to 0} class.\ error(\mathcal{F}_1) \approx class.\ error(f_0) \tag{3}$$

This is a natural consequence of the fact that instead of observing a data packet's delivery in promiscuous mode by the node s_i, it can be equally well done in a cooperative way by s_i and s_{i+2}, if the window size (sampling frequency) is small. In other words, if the time window is small enough that it always includes only a single event, the relationship between events at s_i and s_{i+2} becomes explicit. This is however connected with a very high communication cost (each packet arrival at s_{i+2} must be explicitely reported to s_i). The following observations can be formulated:

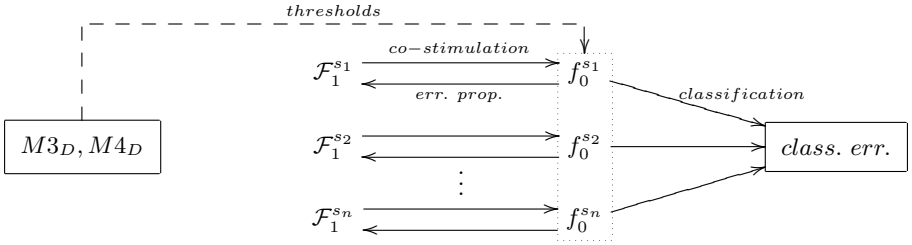
- If using f_0, the thresholds for the watchdog features can be set directly, for example M3 could be set to 0.90 (90% data packets must be correctly forwarded) in order to classify s_{i+1} as misbehavior free.
- If using $\mathcal{F}_1$ with $win.\ size \gg 0$, learning based on data traffic at both s_i and s_{i+2} must be done. Feature averaging over a time window increases the classification task complexity. Frequency of the extra communication between s_i and s_{i+2} depends on the time window size.

Considering the above two observations and Eq. 3, we propose the following algorithm.

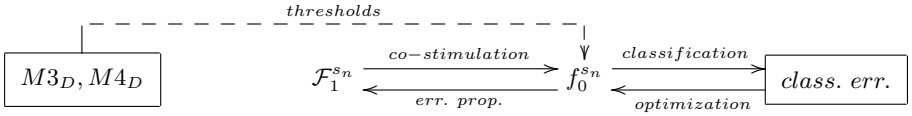
Error Propagation Algorithm:

1. Choose desired levels (thresholds) for the forwarding and processing delay indexes denoted as $\mathrm{M3}_D$ and $\mathrm{M4}_D$, respectively.
2. The network starts operating. Allow all nodes to use promiscuous mode. Each node builds f_0 and $\mathcal{F}_1$ sample sets. Disable promiscuous mode at a node, when the sample set size reaches a target value.
3. Label the $\mathcal{F}_1$ based samples. A $\mathcal{F}_1$ based sample will be labeled as "normal", if in the f_0 based sample from the corresponding time window, $M3 \geq M3_D$ and $M4 \leq M4_D$, and as "misbehavior" otherwise.
4. Error propagation: compute a classifier by using only the samples based on $\mathcal{F}_1$.
5. Use the $\mathcal{F}_1$ based classifier computed in the above step to classify any fresh $\mathcal{F}_1$ samples. Co-stimulation: if a sample gets classified as "misbehavior", allow s_i to compute a f_0 sample in promiscuous node.
6. Classification: apply the $\mathrm{M3}_D$ and $\mathrm{M4}_D$ thresholds to the fresh f_0 based sample. If the "misbehavior" classification gets confirmed, mark s_{i+1} as misbehaving. Normal behavior can get classified in a similar way.

The algorithm is schematically depicted in Fig. 2(a). The error propagation step utilizes the rule described in Eq. 3. The classification error induced by



(a) An approach with error propagation and co-stimulation.



(b) An approach extended with optimization.

Fig. 2. Error propagation algorithm.

the $M3_D$ and $M4_D$ threshold values gets this way propagated onto the $\mathcal{F}_1$ feature set. Since for practical reasons $win.\ size \gg 0$ must hold, a co-stimulation ($\mathcal{F}_1 \longrightarrow f_0$) is necessary for improving the classification performance. The less energy effcient f_0 based classification is thus used only if a co-stimulation from its $\mathcal{F}_1$ based counterpart is present. In other words, any detection based on $\mathcal{F}_1$ (adaptive immunity) must coincide with some damage being explicitely detected. This damage detection is based on f_0 (innate immunity).

The rule in Eq. 3 implicates that the final classification performance should equal the classification ability of the two watchdog features M3 and M4, if a small time window is applied. This implies that the values for $M3_D$ and $M4_D$ must be reasonably chosen in order to minimize the classification error.

The error propagation algorithm can be extended with an optimization phase; see Fig. 2(b). The classification outcome can be used to find new thresholds for the f_0 based samples. In this case, the threshold values $M3_D, M4_D$ serve as seed values for the whole process. Then it is possible to relabel the $\mathcal{F}_1$ samples and to recompute the related classifier. Co-stimulation and classification phases are then executed. In general, any suitable optimization procedure with several optimization and error propagation cycles could be applied. In order to use this extended approach in a distributed way, an estimation of natural packet losses (noise) at nodes must be done.

As shown in Fig. 1(d), any $\mathcal{F}_1$ computation is also based on the information available at s_{i+2}. This two-hop dependency can be compared to synapses among neurons. Under this comparison, our extended approach bears a certain similarity to the *backpropagation algorithm* for artificial neural networks [17].

7 Experimental Setup

The network simulation was done with the JiST/SWANS [18] network simulator.

Topology, Connections, Data Traffic and Protocols: We used a topology based on a snapshot from the movement prescribed by the Random waypoint movement model [19]. There were 1,718 nodes simulated; the physical area size was 3,000m × 3,000m. We used this topology because it has been quite extensively studied by Barrett et al. in [20]; results reported therein include many graph-theoretical measures that were helpful in finding suitable parameters for our experiments.

We modeled data traffic as Constant bit rate (CBR), i.e. there was a constant delay when injecting data packets. This constant delay in our experiments was 2 seconds (injection rate of 0.5 packet/s); the packet size was 68 bytes. CBR data packet sources correspond e.g. to sensors that transmit their measurements in predefined constant intervals. CBR can be considered due to its synchronized nature an extreme model for data packet injection. In fact, the results published in [21] show that when using a stochastic injection model, such as the Poisson traffic model, one can expect a better performance of the detection system.

We used 50 concurrent connections. The connection length was 7 hops. In order to represent a dynamically changing system, we allowed connections to expire. An expired connection was replaced by another connection starting at a random source node that was not used previously. Each connection was scheduled to exist approximately 15 to 20 minutes. The exact connection duration was computed as

$$\delta + r_U \lambda \tag{4}$$

where δ the desired duration time of a connection, r_U a random number from the uniform distribution $[0, 1]$ and λ the desired variance of the connection duration. In our experiments, we used $\delta = 15\ min$ and $\lambda = 5\ min$.

We used the AODV routing protocol, IEEE 802.11b MAC protocol, UDP transport protocol and IPv4. The channel frequency was set to 2.4 GHz. The bandwidth was set to 2 Mbps. Antenna and signal propagation properties were set so that the resulting radio radius equals 100 meters.

Misbehavior: There were 20 wormholes in each simulation run; each wormhole was designed to bypass 15 hops, i.e. the source and sink were 15 hops away before a given wormhole was activated. There were 236 nodes randomly chosen to execute DATA dropping or delaying misbehavior. Our intention was to model *random failure occurrences*, assuming a uniform failure distribution in the network. As it is hard to predict the routing of packets, many of these nodes could not execute any misbehavior as there were no DATA packets to be forwarded by them. In our case, about 20-30 misbehaving nodes were concurrently active.

Experiments: We did 20 independent runs for each misbehavior type and 20 misbehavior free (normal) runs. The simulation time for each run was 4 hours. We used a non-overlapping time window approach for features' computation. We used four different time window sizes: 50, 100, 250 and 500 seconds. In case of a 500-second time window, there were 28 non-overlapping windows in each run (4

	Normal				Misbehavior			
Win. size[s]	Det. rate	$CI_{95\%}$	FP rate	$CI_{95\%}$	Det. rate	$CI_{95\%}$	FP rate	$CI_{95\%}$
	f_0 (M3$_D$ and M4$_D$)							
50	96.15	1.96	0.46	0.02	99.25	0.05	6.36	7.01
100	96.25	2.01	0.20	0.01	99.68	0.03	6.16	7.10
250	96.17	2.11	0.15	0.01	99.76	0.03	6.28	7.33
500	95.84	2.17	0.12	0.01	99.81	0.04	6.71	7.20
	$\mathcal{F}_1$							
50	93.56	2.37	2.09	0.20	96.54	0.61	10.59	9.08
100	92.38	2.64	2.81	0.49	95.27	1.71	12.45	9.68
250	89.05	2.91	5.04	0.90	91.55	2.29	17.97	11.78
500	86.10	3.93	7.49	1.60	87.69	1.78	22.36	11.94
	$\mathcal{F}_1$ and then f_0 (M3$_D$ and M4$_D$)							
50	93.23	2.40	0.12	0.01	95.48	0.89	5.41	5.67
100	92.24	2.66	0.05	0.00	94.24	2.44	5.24	6.02
250	88.97	3.00	0.06	0.00	90.35	3.00	5.23	6.10
500	86.04	3.95	0.06	0.01	85.51	2.39	4.84	5.31

Table 1. Performance. $M3_D = 0.975$, $M4_D = 4$ms.

hours/500 seconds = 28.8). This gave us $5 \times 28 \times 20 = 2{,}800$ vectors (samples) for each node. This also determined the max. target sample size for Step 2 of our algorithm.

$M3_D$ and $M4_D$ threshold values: We set M3$_D$ to 97.5% and M4$_D$ to 4ms. These values were found through the extended approach shown in Fig. 2(b) using the "trial and error" method (as a substitute for a formal optimization approach).

Induction algorithm: We used a decision tree classifier. To decide whether a node within the decision tree should be further split (impurity measure), we used the information gain measure [17]. As the decision tree classifier is a well-known algorithm, we omit its discussion. We refer the interested reader to [17]. We used the decision tree implementation from the Rapidminer tool [22].

The estimation of classification performance was done using a stratified k-fold cross-validation approach [17] with $k = 20$.

8 Performance Evaluation

The results achieved by applying the error propagation algorithm are reported in Table 1. It can be seen that as the time window size decreases, the performance of f_0 and $\mathcal{F}_1$ becomes more and more comparable. It can be also seen that the final co-stimulation with f_0 improves the performance only in a limited way compared to the f_0 only approach. More specifically, the FP rate for "normal" behavior could be significantly decreased. The confidence intervals ($CI_{95\%}$) belonging to the FP rates for "normal" behavior are almost zero. The FP and $CI_{95\%}$ rates

for "misbehavior" could not be significantly decreased compared to f_0. This performance gap between "normal" and "misbehavior" originates from the fact that learning "normal" is simpler than learning "misbehavior". In the latter case, samples which belong to "normal" are often misinterpreted as misbehaving. This is a result of the unsophisticated initial sample separation (labeling) based only on the two watchdog features. This increases the FP rate and the $CI_{95\%}$ of the misbehavior class.

The average sample size per node was 6,941, if the window size was set to 50 seconds. As not every node was continuously involved in data packet forwarding, the sample size was lower than the theoretical max. size.

Notice that according to our approach, when detecting a misbehavior, the more costly f_0 based detection will only get used, if (i) a true positive was detected by $\mathcal{F}_1$ or (ii) a false positive was mistakenly detected by $\mathcal{F}_1$. This means, for a misbehavior free ad hoc network, the energy saving over an exclusive f_0 approach is proportional to $1 - FP\ rate$, where $FP\ rate$ is in this case the $\mathcal{F}_1$ based false positives rate for the misbehavior class.

9 Conclusions

We proposed and tested an approach inspired by the role of co-stimulation in the BIS. Our preliminary results show that this approach has a positive effect on both energy efficiency of misbehavior detection and the false positives rate. The achieved detection rate was in the $86 - 95\%$ range depending on the monitoring window size. Our error propagation algorithm can be used for both misbehavior detection as well as for checking whether an ad hoc network complies with its operational strategy. We pointed out that our approach can be extended with an optimization technique that would allow for a classification error minimization.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under the grant no. SZ 51/24-2 (Survivable Ad Hoc Networks – SANE).

References

1. Yegneswaran, V., Barford, P., Ullrich, J.: Internet intrusions: global characteristics and prevalence. Proc. of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (2003) 138–147
2. Anantvalee, T., Wu, J.: A survey on intrusion detection in mobile ad hoc networks. Wireless/Mobile Network Security (2007) 159–180
3. Murphy, K., Travers, P., Walport, M.: Janeway's immunobiology. Garland Pub (2008)
4. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The Link between AIS and IDS? Proc. of International Conference on Artificial Immune Systems (ICARIS) (2003) 147–155

5. Hofmeyr, S., Forrest, S.: Immunity by design: An artificial immune system. Proc. of Genetic and Evolutionary Computation Conference (GECCO) **2** (1999) 1289–1296
6. Sarafijanovic, S., Le Boudec, J.: An artificial immune system for misbehavior detection in mobile ad-hoc networks with virtual thymus, clustering, danger signal and memory detectors. Proc. of International Conference on Artificial Immune Systems (ICARIS) (2004) 342–356
7. Drozda, M., Schaust, S., Szczerbicka, H.: AIS for Misbehavior Detection in Wireless Sensor Networks: Performance and Design Principles. Proc. IEEE Congress on Evolutionary Computation (CEC) (2007) 3719–3726
8. D'haeseleer, P., Forrest, S., Helman, P.: An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. IEEE Symposium on Security and Privacy (1996) 110–119
9. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. Theor. Comput. Sci. **403**(1) (2008) 11–32
10. Kim, J., Bentley, P., Wallenta, C., Ahmed, M., Hailes, S.: Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm. Proc. of International Conference on Artificial Immune Systems (ICARIS) (2006) 390–403
11. Drozda, M., Schildt, S., Schaust, S.: An Immuno-Inspired Approach to Fault and Misbehavior Detection in Ad Hoc Wireless Networks. Technical report, Leibniz University of Hannover (2009)
12. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **97**(1-2) (1997) 273–324
13. Perkins, C.E., Royer., E.M.: Ad hoc On-Demand Distance Vector Routing. Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (1999) 90–100
14. Feeney, L., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. Proc. of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM **3** (2001)
15. ZigBee Alliance: ZigBee Specification. (2005)
16. Hu, Y., Perrig, A., Johnson, D.: Wormhole attacks in wireless networks. IEEE Journal on Selected Areas in Communications **24**(2) (2006) 370–380
17. Alpaydin, E.: Introduction To Machine Learning. MIT Press (2004)
18. Barr, R., Haas, Z., van Renesse, R.: JiST: an efficient approach to simulation using virtual machines. Software Practice and Experience **35**(6) (2005) 539–576
19. Johnson, D., Maltz, D.: Dynamic source routing in ad hoc wireless networks. Mobile Computing **353** (1996) 153–181
20. Barrett, C., Drozda, M., Engelhart, D., Kumar, V., Marathe, M., Morin, M., Ravi, S., Smith, J.: Understanding protocol performance and robustness of ad hoc networks through structural analysis. Proc. of the IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob'2005) **3** (2005) 65–72
21. Schaust, S., Drozda, M.: Influence of Network Payload and Traffic Models on the Detection Performance of AIS. Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'08) (2008) 44–51
22. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (2006) 935–940