Costimulation and Priming: Can it Help Protect Ad Hoc Wireless Networks?

Martin Drozda and Sven Schaust
Simulation and Modeling Group, Faculty of Electrical Engineering and Computer Science, Leibniz University of Hannover
Welfengarten 1, 30167 Hannover, Germany.
Email: {drozda,svs}@sim.uni-hannover.de

***Abstract*—We review two key mechanisms of the Biological immune system (BIS): costimulation and priming. Then we explain the relevance of these two mechanisms for misbehavior detection in ad hoc wireless networks. We argue that costimulation and priming can not only increase the reliability and robustness of misbehavior detection but also help increase the energy efficiency. We conclude by giving an outlook on future research related to the design of security protocols inspired by the efficiency of the BIS.**

I. Introduction

The Biological immune system (BIS) [1] can efficiently protect its host against microorganisms such as viruses, bacteria, fungi or parasites. The BIS consists of two integral parts: the innate and the adaptive immune system. Whereas the innate immune system can be found in living organisms ranging from plants to vertebrates, the adaptive immune system can only be found in more complex organisms such as jawed vertebrates. These two parts of the BIS act in an orchestrated way when an elimination of foreign microorganisms, or pathogens, is necessary.

The undisputed efficiency of the BIS got quickly recognized as a potent source of inspiration for the design of architectures and methods aimed at protection of technical systems. Artificial immune systems (AIS) [2], the technical counterpart of the BIS, emerged as a well recognized research area within the field of computational intelligence.

Ad hoc wireless networks [3] can be described as an extreme form of wireless communication. They are based on wireless devices that, in many scenarios, are expected to have limited energy, computation and memory resources. Additionally, they do not rely on any form of fixed infrastructure such as wireline or base stations. Instead data packet routing is done by any member of the network. Due to their infrastructureless nature, ad hoc wireless networks are easy to deploy. This comes however at an increased maintenance complexity.

Due to these restrictions, much research effort in the area of ad hoc wireless networks was devoted to the design of self-tuning communications protocols. In the following, we concentrate on security aspects of protocols that benefit from two basic BIS mechanisms: costimulation and priming.

This document is organized as follows. In Sec. II we present the BIS and its mechanisms. In Sec. III we take a closer look at the immunology of costimulation and priming. We also formulate their computational interpretations. In Sec. IV we review costimulation approaches know from literature. In Sec. V we review a priming approach for inducing misbehavior resistance in ad hoc wireless networks. In Sec. VI we provide a further discussion of the reviewed approaches and sum up the related research challenges. In Sec. VII we conclude.

II. The Biological Immune System

The Biological immune system is remarkably efficient in correctly detecting and eliminating pathogens, and in choosing the correct immune response. When confronted with a pathogen, the BIS relies on the 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.

For an immune reaction to occur, it is often necessary that first, a cell has been classified as a pathogen and second, this cell could cause some damage to the human organism. This means that the BIS is only reactive with *infectious* cells, i.e. with pathogens that can indeed cause harm [1]. The necessity of such an interplay, or *costimulation*, within the innate and adaptive immune systems suggests that a two-way communication among various immune cells is a common phenomenon.

To understand the interaction between these two parts of the immune system, we briefly introduce several immune cell types. T-cells and B-cells (or lymphocytes) play a central role in the adaptive immune system. T-cells are able to recognize antigen fragments produced by specialized antigen presenting cells (APCs). Antigen is a substance that can be found on the surface of a pathogen. APCs are cells able to display antigen fragments on their surface; they are also able to preprocess an antigen so that a detection by the adaptive immune system is possible. The first encounter of a T-cell with an antigen is called *priming*. Through priming, a naive T-cell becomes activated. When a pathogen recognition by a T-cell occurs, a signaling process, that mobilizes other players of the BIS, is performed.

Unlike T-cells, B-cells are able to recognize pathogens without the extra help of an APC. If a recognition happens, the B-cell gives rise to many plasma cells producing *antibodies*. Antibodies are able to mark a cell or microorganism as pathogen. This significantly streamlines the elimination.

The role of Dendritic cells (DCs) [4] is to induce immunological tolerance, and activate and regulate the adaptive system. DCs exist in several stages. In the precursor stage, they are able to secrete inflammatory or antiviral *cytokines*. Secretion of cytokines is a strong signal that a harm causing process is underway in the organisms. Immature DCs can capture and analyze a pathogen. Upon contact with the pathogen, they can differentiate into either an APC or a mature DC. They have a unique ability to recognize tissue injury, process signals from other immune cells (including DCs in the precursor or immature stage) and alert the adaptive immune system. DCs thus provide a link between innate and adaptive immunity. The following patterns and signals are known to play a major role in their activation:

- PAMP: Pathogen-associated molecular patterns describe molecules which are common in specific groups of pathogens (e.g bacteria).
- Danger signal is emitted as the result of an unexpected cell death. The molecules representing a danger signal accumulate in the tissue, causing a reaction of dendritic cells as they are sensitive to the signal concentration.
- Safe signal is emitted as the result of an expected cell death. A safe signal therefore suppresses immune reactions.
- Inflammatory signal is released when a tissue injury occurs. Generally the process of inflammation is not enough to fully activate a dendritic cell alone, but it supports its reaction towards the other three signals.

The ability of the DCs to recognize harm has been induced over the evolutionary time and is a part of e.g. the human genetic information.

III. Costimulation and Priming: A Close-up Perspective View

A. Costimulation

In the recent years, costimulation has received an increased attention from immunologists. There is a large body of scientific evidence about the fact that several signals, besides the ones sent by the antigen receptor, are required for a full activation of a lymphocyte. Without these additional signals, lymphocytes are unable to proliferate, become non-responsive and can even die [1]. According to Frauwirth and Thompson [5], it has become clear that the interaction between receptor/ligand pairs on a T-cell and an APC represents a critical event in the activation process. A similar process is applied in case of B-cell activation. It is this event that is generally referred to as costimulation in immunology.

Costimulation can be thus defined as the involvement of *"reciprocal and sequential signals between cells"* in order to fully activate a lymphocyte [5].

Within this process, an interaction between T-cells and APCs begins when the T-cell antigen receptor is stimulated by a specific major histocompatibility complex (MHC) on the surface of the APC. Essential concentrations of specific costimulatory molecules (B7) on the APC activate a specific group of T-cell surface molecules (CD28).[1] This then adjusts

[1]CD = Cluster of Differentiation.

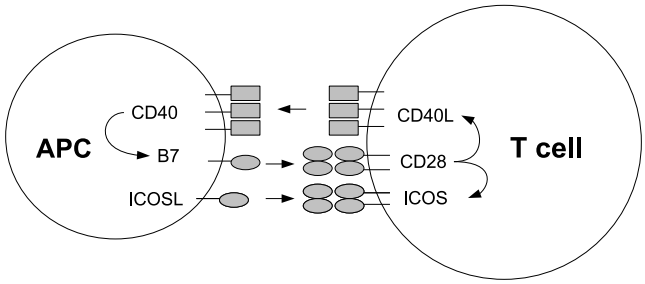


Fig. 1. Costimulation of a T-cell

the level of CD40L. CD40L then binds to another specific antigen on the APC, further increasing the B7 level and boosting the CD28/CD40 positive feedback loop.

The CD28 costimulation also activates the T-cell expression of inducible costimulatory proteins (ICOS), allowing a second level of costimulation by APC-expressed ligand of ICOS (ICOSL). Other costimulatory and inhibitory molecules, which are regulated by the initial costimulatory signals, can further shape the specific outcome of the interaction process. This costimulation process is shown in Fig. 1.

B. Priming

Priming in the BIS describes the effects of the first encounter of an antigen by a naive T- or B-cell. More specifically, immunologists define priming as the activation and clonal expansion of naive T- or B-cells into effector cells after the first encounter with a specific antigen. Such activated cells are then able to induce a full immune response.

Priming can be however also understood in a more general way. For example in the context of plant immunity, immunologist talk about priming plants with resistance against certain diseases [6].

C. Costimulation and Priming: A Computational Interpretation

Before discussing the different approaches which take advantage of costimulation and priming, we define costimulation and priming in more general way, so that a computational interpretation with focus on system security is straightforward.

Definition Costimulation is an auxiliary signal confirming that a misbehavior causes harm to the system.

Definition Priming is a process for specifying the alertness levels of a system with respect to a misbehavior class.

Notice that costimulation is, with respect to the above definition, an auxiliary mechanism to be used in conjunction with another detection mechanism.

Further Definitions

Before we proceed any further, we define and explain several relevant terms used frequently hereafter.

Each node (wireless device) in an ad hoc wireless network has the ability to observe a variety of protocol states and events. Based on these states and events, performance measures or *features* can be computed. *Watchdogs* are a

specific type of features that require promiscuous mode. In *promiscuous mode*, a node listens to the on-going traffic among other nodes in the neighborhood and collects information from the overheard packets.

Misbehavior can be the result of both an intrusion, and a software or hardware failure. In the former case, a wireless device can become fully controlled by an attacker. In the latter case, misbehavior is rather a consequence of software or hardware design flaws.

Classification is understood as class prediction for a vector of features. In misbehavior detection, we deal with the normal class, representing the usual network behavior, and several misbehavior classes. Classification error is a measure that captures the misclassification rate of a given classification approach applied in the classification of a set of feature vectors.

IV. Costimulation Approaches

A. Early Costimulation Approaches

The early 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 [7] described an AIS able to detect anomalies in a wired TCP/IP network. Costimulation was in their setup done by a human operator who was given 24 hours to confirm a detected attack.

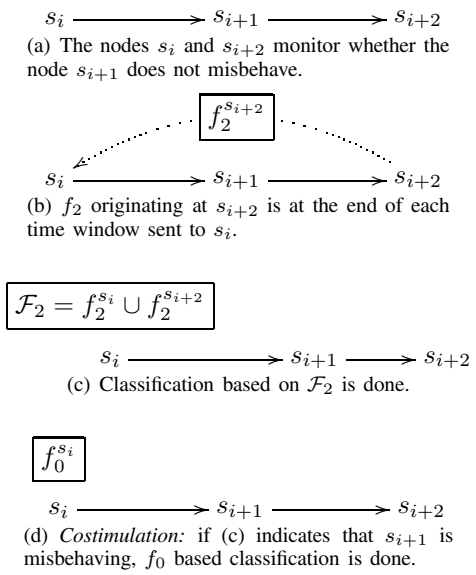
Sarafijanović and Le Boudec [8] introduced an AIS for misbehavior detection in mobile ad hoc wireless networks. They used four different features based on the network layer of the OSI protocol stack. A costimulation in the form of a danger signal emitted by a connection source was used to inform nodes on the forwarding path about perceived data packet loss. Such a signal could then be correlated with local detection results.

The costimulation approach of Hofmeyr and Forrest is a direct implementation of an auxiliary signal produced by a human operator. The approach of Sarafijanović and Le Boudec is tightly coupled with TCP (Transmission control protocol). It is thus also not general enough to be applied to an arbitrary ad hoc wireless network. Many types of ad hoc wireless networks (such as sensor networks) are expected to work with a reduced service set at the transport layer of the OSI protocol stack. This precludes the use of many key TCP services.

B. Dendritic Cell Algorithm

The Dendritic Cell Algorithm (DCA) was introduced by Greensmith et al. [9]. The DCA is an agent-based classification algorithm that uses a fixed population and a signal matrix with a weighted sum equation. The signal matrix is based on pre-categorized signal types inspired by the signal processing capabilities of dendritic cells. These signals are: PAMP signal, danger signal, safe signal and inflammatory (amplifying) signal.

Based on these signals and their weights in the equation, three output signal values of a dendritic cell are computed. Depending on this output signal values, a maturity stage is assigned to the dendritic cell. Each dendritic cell can be either in costimulatory, semi-mature or mature phase. A semi-mature DC provides a tolerance stimulation (observed data is not harmful), while a mature DC provides a reactive stimulation (observed data is harmful).



(a) The nodes s_i and s_{i+2} monitor whether the node s_{i+1} does not misbehave.

(b) f_2 originating at s_{i+2} is at the end of each time window sent to s_i.

(c) Classification based on $\mathcal{F}_2$ is done.

(d) *Costimulation:* if (c) indicates that s_{i+1} is misbehaving, f_0 based classification is done.

Fig. 2. A costimulation model for ad hoc networks.

Any data (feature vector) subject to classification is presented to a subset of dendritic cells from the population pool. If a significant number of dendritic cells is driven into the mature stage, the data is classified as representing an anomaly (misbehavior).

An anomaly detection approach based on the DCA was proposed by Kim et al. in [10]. Several types of signals, each having a different function are employed in order to detect routing manipulation in sensor wireless networks.

Since the DCA is a general purpose classification algorithm, it can be used as the underlying classifier in misbehavior detection. Stibor at el. [11] pointed out recently that the DCA is a linear classifier with a limited classification power.

C. Energy Efficient Costimulation Approach

Recently, Drozda et al. proposed a costimulation inspired approach to misbehavior detection in ad hoc wireless networks [12]. The goal was to offer a tradeoff between detection performance and energy consumption. Two qualitatively different feature sets $\mathcal{F}_2$ and f_0 were identified. The features in these sets are averaged over a time window of size $win.\ size$. The properties of $\mathcal{F}_2$ and f_0, with respect to the classification error ϵ and energy efficiency ξ, can be summarized as follows:

1) For $win.\ size \to 0$, it holds:
$$\lim_{win.\ size \to 0} \epsilon(\mathcal{F}_2) \approx \epsilon(f_0) \quad (1)$$
This implies that the classification precision based on these two sets equalizes, if a small time window size is applied.
2) For $win.\ size \gg 0$, it holds:
$$\epsilon(\mathcal{F}_2) > \epsilon(f_0) \quad (2)$$

$$\xi(\mathcal{F}_2) > \xi(f_0) \tag{3}$$

The measure of energy efficiency ξ includes feature computation costs as well as all induced communication costs.

The disadvantage of $\mathcal{F}_2$ based classification is its lesser classification performance, if compared to f_0, for $win.\ size \gg 0$. On the other hand, it is significantly more energy efficient.

For details on these two features sets, we refer the reader to [12]. The features in $\mathcal{F}_2$ and f_0 were identified using a wrapper approach due to John and Kohavi [13].

The approach is based on a two-stage classification. First, a preliminary classification is done with an energy efficient feature set $\mathcal{F}_2$. If an anomalous behavior is detected, a more precise but more energy demanding classification is performed with f_0. *Costimulation* is thus in this setup a conditional use of f_0 based classification with respect to the outcome of $\mathcal{F}_2$ based classification. Both classification steps assume a supervised learning algorithm – a decision tree classifier was used in [12]. In other words, the $\mathcal{F}_2$ based classification provides an auxiliary signal for the f_0 based classification:

$$\mathcal{F}_2 \xrightarrow{costimulation} f_0 \tag{4}$$

This costimulation approach is illustrated in Fig. 2. The nodes s_i, s_{i+1} and s_{i+2} are part of a data flow. The node s_{i+1} is the node being monitored for misbehavior. The feature subsets included in $\mathcal{F}_2$, $f_2^{s_i}$ and $f_2^{s_{i+2}}$, are computed by the nodes s_i and s_{i+2}, respectively. The features $f_2^{s_{i+2}}$ computed by s_{i+2} are sent at the end of *each time window* to s_i. Then an $\mathcal{F}_2$ based classification is done. f_0 is computed by s_i in promiscuous mode.

Let α_2 and β_2 be the detection and false positives rate, respectively, associated with $\epsilon(\mathcal{F}_2)$ with respect to a considered class.[2] Similarly, let α_0 and β_0 be the detection and false positives rate, respectively, associated with $\epsilon(f_0)$.

Observation 1 For $win.\ size \gg 0$, with respect to Eq. 1, it holds that $\alpha_2 < \alpha_0$ and $\beta_2 > \beta_0$.

Since however, the f_0 based classification is used conditionally with regards to the outcome of the $\mathcal{F}_2$ based classification, for the final detection rate α and final false positives rate β observed after both classification stages, the following observation can be formulated:

Observation 2 For $win.\ size \gg 0$, it holds that $\alpha \leq \alpha_2$. It also holds, $\beta = \beta_0$.

This means, the final detection rate has an upper bound determined by the $\mathcal{F}_2$ based classification. The final false positives rate is only depending on the f_0 based classification.

Let us now illustrate the above two observations through an example. In [12], it was reported that for a misbehavior class, the time window of 500 seconds and with respect to the used experimental setup, $\alpha_2 = 78.93\%$, $\beta_2 = 26.97\%$, $\alpha_0 = 98.45\%$ and $\beta_0 = 2.92\%$. The final detection and false positives rates were $\alpha = 73.99\%$, $\beta = 2.44\%$, respectively. β_0 is sligtly higher than β but this difeference was not found to be statistically significant.

In a misbehavior free network, the rate at which the f_0 based classification gets mistankenly applied is related to β_2, the false positives rate of the $\mathcal{F}_2$ based classification. It was demonstrated that for this particular setting about 97.8% energy resources could be saved.

V. Towards Priming against Misbehavior

Drozda et al. proposed an algorithm for enforcing a network operational strategy in an *energy efficient* way [14]. This algorithm builds upon Eq. 1. Unlike in the costimulation approach, where two independent classifiers based on f_0 and $\mathcal{F}_2$ must be computed, only the computation of an $\mathcal{F}_2$ based classifier is necessary. This is achieved through the definition of priming thresholds. These thresholds determine the basis for the alertness of a given network with respect to a misbehavior type. A priming threshold can be e.g. the maximum allowed data packet loss at a node or the maximum allowed data packet processing delay at a node. Ths implies, misbehavior is defined as a violation of the priming thresholds $\mathcal{P} = \{p_1, p_2, ..., p_l\}$, where l is the number of priming thresholds.

Since it is possible to choose f_0 features that can directly measure whether a priming condition is met e.g. a maximum allowed data packet loss for the node s_{i+1}, it is possible to use this information to label a feature vector in $\mathcal{F}_2$ from a corresponding time window. This in turn allows for the computation of an $\mathcal{F}_2$ based classifier. Notice that for $win.\ size \to 0$ with respect to Eq. 1, the classification error associated with such an $\mathcal{F}_2$ based classifier is determined by the choice of $\mathcal{P}$. The classification error induced by the choice of $\mathcal{P}$ gets thus propagated onto the $\mathcal{F}_2$ based classifier.

Since however, for practical reasons $win.\ size \gg 0$, with regards to Eq. 2, a costimulation must follow:

$$\mathcal{F}_2 \underset{error\ propagation}{\overset{costimulation}{\rightleftarrows}} f_0 \begin{array}{l} \nearrow \epsilon \\ \searrow \mathcal{P} \end{array} \tag{5}$$

Costimulation by the $\mathcal{F}_2$ based classification results in the computation of a fresh f_0 based feature vector by s_i. This vectors is then checked for compliance with the priming requirements $\mathcal{P}$.

Let α_2, β_2, α_0, β_0, α and β with respect to a class be defined as in the previous section. The following observations can be formulated:

Observation 3 After the error propagation phase, $f_0 \longrightarrow \mathcal{F}_2$, for $win.\ size \gg 0$, it holds that $\alpha_2 < \alpha_0$ and $\beta_2 > \beta_0$.

Observation 4 After the costimulation phase, $\mathcal{F}_2 \longrightarrow f_0$, for $win.\ size \gg 0$, it holds that $\alpha \leq \alpha_2$ and $\beta = \beta_0$.

What makes this priming approach distinct from the costimulation approach, discussed in the previous section, is the origin of the f_0 related classification error. In the costimulation approach, the classification error is determined by the f_0 based

[2]Classification error is a total measure of misclassification. Detection rate and false positives rate are measures of misclassification with respect to a given class. In case of misbehavior detection, the focus is on the misbehavior class.

classifier and its inputs (including features). In the priming approach, the classification error is determined by the choice of thresholds $\mathcal{P}$ for priming. For example, the maximum allowed data packet loss can be set to 2.5%. If a node drops more data packets, it will be classified as misbehaving. In an ad hoc network, a 2.5% data packet loss can be however caused by medium contention, corrupt data packets or stale routes. In this case, a node would get incorrectly classified as misbehaving, hence the classification error.

Error propagation and costimulation is executed within the same node; see Fig. 3. The priming thresholds $\mathcal{P}$ are used for the labeling of $\mathcal{F}_2$ feature vectors as well as when inspecting the fresh f_0 based feature sample. Notice that some nodes in the example network, for the shown data flows, are unable to compute $\mathcal{F}_2$ since they have no two-hop neighbor s_{i+2}. On the other hand, several nodes receive multiple $f_2^{s_{i+2}}$, e.g. s_1 from s_3 and s_5.

In [14], priming with respect to two priming thresholds was done. The following results were reported for a similar experimental setup (other nodes were used) that was used for testing of the costimulation approach: $\alpha_0 = 99.81\%$, $\beta_0 = 6.71\%$, $\alpha_2 = 87.69\%$ and $\beta_2 = 22.36\%$. The final detection and false positives rates were $\alpha = 85.51\%$, $\beta = 4.84\%$, respectively. As in the case of costimulation, it can be seen that $\alpha < \alpha_2$ and $\beta \cong \beta_0$. The rather high final false positives rate is caused by applying identical priming thresholds to all nodes in a $1,718$-node ad hoc network. The arguments for energy efficiency of this priming approach are identical to those stated for the costimulation approach.

With respect to the final classification outcome, the individual priming threshold values for each node can be optimized, i.e. the classification error can be minimized. This can be done by a repeated application of the error propagation and costimulation phases, while adjusting the priming thresholds for each node, until a termination condition is met; see Fig. 4. This approach can be described as priming with influences of e.g. noise being locally considered. More formally, new optimized priming thresholds $\mathcal{P}^* = \{p_1^*, p_2^*, ..., p_l^*\}$ for each node s_i will be found, so that the final classification error for each node s_i is minimized.

Priming with optimization is distantly similar to the *back-propagation algorithm* for artificial neural networks [15]. Unlike in the backpropagation algorithm where edge weights must be recomputed in order to reflect the classification error, here the adjustment is done by a repeated computation of the classifier $\mathcal{F}_2$ at each node (wireless device). Synoptic connections among nodes are provided by the transmitted $f_2^{s_{i+2}}$ feature sets.

VI. Discussion and Future Challenges

Since the early experimentation with costimulation due to Forrest et al., the mechanism has been gaining attention in the AIS community. The algorithms presented in Sec. IV are capable of adapting to different environments, especially if combined with optimization and learning strategies.

Translation of the priming principle to the field of misbehavior detection is relatively new and has been done only recently by Drozda et al. in [14]. Priming a protection system

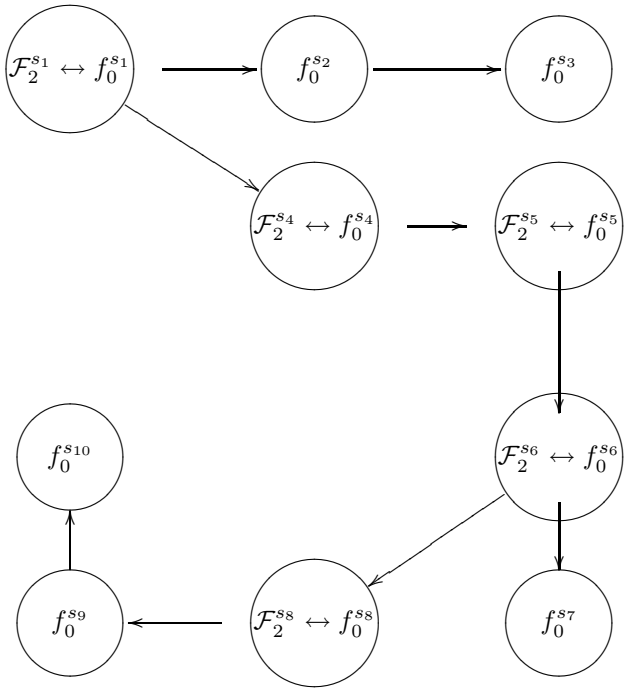


Fig. 3. A 10-node ad hoc network with priming. s_1 is the only data flow source node. s_3, s_7 and s_{10} are sink nodes for three distinct data flows.

is vital to making its responses to various misbehavior classes predictable.

In order to further advance these immuno-inspired approaches, several challenges must be addressed:

1) *Protection = detection + response:* Common to all approaches discussed herein is that they do not offer any immuno-inspired remedy for dealing with already detected misbehaving nodes, i.e. any node "elimination" relies on the usual protocol procedures such as removing it from routing tables.
2) *Identifying the costimulation source:* The results reported by Sarafijanović and Le Boudec [8] suggest that the transport layer can serve as a potent source of costimulation for any lower layer detection mechanisms. This is however not always possible, since e.g. sensor networks are expected to operate with a reduced set of transport layer services. Therefore a distinctly different form of costimulation was necessary [12].
3) *Detection rate vs false positives rate split:* The costimulation and priming approaches reported in [12], [14] allow that the final detection rate and false positives rate are influenced by two distinct mechanisms. This in turn allows for some freedom when the misbehavior detection system needs to be tuned with respect to these two measures in isolation. It is however an open question how to provide some strict detection performance guarantees with respect to a given misbehavior class.
4) *Detection performance vs energy efficiency trade-off:* The costimulation and priming approaches reported

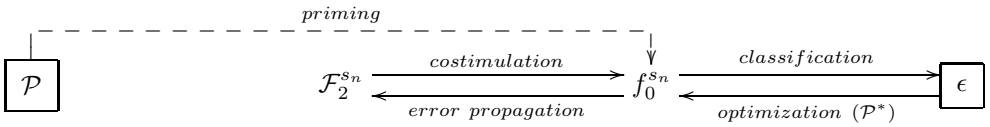


Fig. 4. Priming model.

in [12], [14] also allow for flexibility in choosing the detection and energy efficiency levels. By adjusting the time window size, it is possible to choose a trade-off between these two objectives.

5) *Further considering energy restrictions:* The approaches reported in [12], [14] significantly improve the energy efficiency of misbehavior detection. They however partly rely on features computed in promiscuous mode. Detection mechanisms based on promiscuous mode offer a very good detection performance, but decrease the overall energy efficiency. It is therefore mandatory to search for more sophisticated features which offer a reasonable detection performance and, at the same time, are energy efficient.
6) *Level of abstraction*: Various forms of signals within the BIS contribute to its efficiency and robustness. It is necessary that a translation of these capabilities to a computational paradigm will be completed. The efforts within the DCA, in our opinion, do not capture the fine non-sequential nature of immune mechanisms.
7) *Priming:* Priming allows that detecting a deviation from an operational network strategy becomes straightforward. Some priming variables are easy to identify, e.g. the maximum allowed data packet loss at a node or the maximum allowed data packet processing delay at a node. However, a general method for finding a larger set of priming variables is missing.
8) *Convergence of priming with optimization:* Efficient learning and optimization strategies are necessary to minimize the transient phase of the system after deployment. It is so far not clear which optimization procedure would, in this respect, fit best the priming approach.
9) *Generalization of priming:* Priming could also play a major role in the response phase after a misbehavior is detected, by acting as an enabling mechanism for an active reaction, so that a specific operation strategy can be *actively* enforced.

VII. Conclusions

We reviewed costimulation and priming, two basic mechanisms of the BIS, with respect to their suitability for misbehavior detection in ad hoc wireless networks. The following general conclusions with respect to these two mechanisms can be drawn: (i) they increase significantly the reliability and robustness of misbehavior detection systems, (ii) they increase significantly the energy efficiency of misbehavior detection; it could be demonstrated that $\sim 98\%$ energy resources can be saved by their application, (iii) they offer flexibility in choosing a trade-off between the detection performance and energy efficiency, (iv) they can be adapted to dynamic and noisy environments.

With respect to the above said, we conclude by saying that costimulation and priming are promising mechanisms which help improve the protection of ad hoc networks.

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] K. Murphy, P. Travers, and M. Walport, *Janeway's immunobiology*. Garland Pub, 2008.
[2] D. Dasgupta, "Advances in artificial immune systems," *IEEE Computational Intelligence Magazine*, pp. 40–49, 2006.
[3] C. Perkins, *Ad hoc networking*. Addison-Wesley Longman Publishing Co., 2001.
[4] J. Banchereau, F. Briere, C. Caux, J. Davoust, S. Lebecque, Y. Liu, B. Pulendran, and K. Palucka, "Immunobiology of dendritic cells," *Annual review of immunology*, vol. 18, no. 1, pp. 767–811, 2000.
[5] K. A. Frauwirth and C. B. Thompson, "Activation and inhibition of lymphocytes by costimulation," *The Journal of Clinical Investigation*, vol. 109, no. 3, pp. 295–299, 2 2002. [Online]. Available: http://www.jci.org/articles/view/14941
[6] H. W. Jung, T. J. Tschaplinski, L. Wang, J. Glazebrook, and J. T. Greenberg, "Priming in Systemic Plant Immunity," *Science*, vol. 324, no. 5923, pp. 89–91, 2009.
[7] S. Hofmeyr and S. Forrest, "Immunity by design: An artificial immune system," *Proc. of Genetic and Evolutionary Computation Conference (GECCO)*, vol. 2, pp. 1289–1296, 1999.
[8] S. Sarafijanovic and J. Le Boudec, "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)*, pp. 342–356, 2004.
[9] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection," *Proc. of International Conference on Artificial Immune Systems (ICARIS)*, pp. 153–167, 2005.
[10] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, and S. Hailes, "Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm," *Proc. of International Conference on Artificial Immune Systems (ICARIS)*, pp. 390–403, 2006.
[11] T. Stibor, R. Oates, G. Kendall, and J. Garibaldi, "Geometrical insights into the dendritic cell algorithm," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 2009, pp. 1275–1282.
[12] M. Drozda, S. Schildt, S. Schaust, and H. Szczerbicka, "An Immuno-Inspired Approach to Fault and Misbehavior Detection in Ad Hoc Wireless Networks," *Submitted to IEEE Conference on Computer Communications (Infocom 2010)*, 2010.
[13] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
[14] M. Drozda, S. Schaust, S. Schildt, and H. Szczerbicka, "An Error Propagation Algorithm for Ad Hoc Wireless Networks," *Proc. 8th International Conference on Artificial Immune Systems (ICARIS'09), LNCS*, vol. 5666, pp. 260–273, 2009.
[15] E. Alpaydin, *Introduction To Machine Learning*. MIT Press, 2004.