European Journal of Control

Fundamental Issues in Control

Guest Editors
S.G. Tzafestas
P.J. Antsaklis

hermes

Lavoisier

6-45
Scanned

Volume 13 Number 2–3 2007

European
Journal
of Control

Volume 13
Number 2-3-2007
March - June

Fundamental Issues in Control

Contents

European Journal of Control (2007)13:105–133
© 2007 EUCA
DOI:10.3166/EJC.13.105–133

European Journal of Control

Security and Trust for Wireless Autonomic Networks

Systems and Control Methods

John S. Baras*

Institute for Systems Research, Electrical and Computer Engineering Department and Computer Science Department,
University of Maryland, College Park

We analyze and solve various problems of security, information assurance and trust in dynamic wireless networks. These include detection and defense against attacks, detection of propagating viruses, evaluation of intrusion systems, attacks at the physical, MAC and routing protocols, trust establishment-dynamics-management. We demonstrate persistently that systems and control models and methodologies provide new and powerful techniques to analyze these problems. We describe the use of distributed change detection methods and algorithms for intrusion detection and the use of non-cooperative games for the detection and defense against attacks at all layers. We demonstrate how Bayesian decision theory can be used to evaluate intrusion detection systems and we resolve some key problems in this area. We use game theoretic methods again to develop robust protocols against attacks, including Byzantine ones. We provide an in-depth investigation of trust establishment and computation in such networks. We describe various methods for distributed trust evaluation and the associated trust (and mistrust) 'spreading' dynamics. We investigate rules and policies that establish 'trust-connected' networks using only local interactions, and find the parameters (e.g. topology type) that speed up or slow down this transition. We describe and explain the phase transition phenomena that we have found in these evolutions. We model the interactions among agents as cooperative games and show that trust can encourage agents to collaborate. This leads us to a fundamental analytical technique, constrained coalitional games, that can be used to evaluate tradeoffs in collaborative networks in various areas: communications, sensors, economics, sociology, biology. We also describe a model for trust evaluation that uses pairwise iterated graph games between the agents to create a 'trust reputation' with evolution coupled to the game dynamics. Finally we present a new modeling framework for trust metric evaluation as linear iterations over ordered semirings. This allows us to formulate problems of resilience of trust metrics and trust evaluation to attacks.

Keywords: Autonomic networks; wireless; security; trust; intrusion detection; dynamic games; change detection; semirings

1. Introduction

The proliferation of networked devices and applications, integrates information technology into our everyday environments. Mobile wireless networks have become the largest component for 'last mile' connections to the global Internet, with the number of connected wireless end-users and devices increasing exponentially. The emerging broadband mobile wireless infrastructures (BMWI) are increasingly affecting all aspects of quality of life and work [7]. Indeed BMWI applications include e-commerce, e-government, e-health, e-education, universal service provisioning, pervasive computing and PDAs, sensor

*Correspondence to: John S. Baras, E-mail: baras@isr.umd.edu

Received 20 December 2006; Accepted 20 January 2007.
Recommended by S.G. Tzafestas and P.J. Antsaklis

networks, connectivity with cars-ships-trains-airplanes, intelligent transportation systems, human health monitoring, health care delivery and management, wireless vehicle networks, disaster relief, homeland security, intelligent buildings. These vast application domains and dramatic changes create unique challenges for network design, management and control. For example an intensive effort has been initiated recently by the USA NSF to re-design the Internet [62].

The traditional centralized server-based management can no longer satisfy the requirements of next generation networks, and new concepts of network structure and management have been proposed. For instance, mobile ad hoc networks (MANETs) [43] aim to provide wireless network services without relying on any infrastructure. The wireless mesh networks, which have been implemented by various groups [3,31], are essentially MANETs over a 802.11 wireless LAN, which can be set up with almost 'zero-cost' in highly mobile environments. Another example is peer-to-peer (P2P) networks [38,78], where a large number of data are shared among millions of network users. All the aforementioned new types of networks share a common characteristic: they are distributed and self-organized, thus they are sometimes called *autonomous/autonomic networks* [1] in the literature. Our fundamental view is that such networks and their design, performance evaluation and operation can be best understood as autonomous, distributed, controlled dynamical systems.

An autonomic network is one that is distributed and mostly asynchronous, self-configuring and self-protecting. Such a network requires minimal administration, which mostly only involves policy-level management. All entities in autonomic networks participate in network control through individual interactions. To achieve desired network management goals under such 'anarchy' is not an easy task.

Autonomous networked systems have been studied in various scientific and engineering fields: in collaborative robotics and networked control [2,45,64] with the recent intensive efforts in this area by the systems and control community; in biological systems, where swarms of bacteria, insects and animals yield sophisticated collective behaviors based on simple individual interactions [14,37]; in physics, where a group of particles interact with their neighbors to achieve certain macroscopic properties, c.f. magnetization [63]; in human societies, where sociologists have modeled human interactions and societal structures evolution using iterated games [32]; in economics where economists have analyzed coalitions and network formations using again extensions of iterated games [32,75]. Thus, the models and methods presented here have wide applicability.

Security, authentication and trust are critical concepts for any network. Indeed the system model we currently and rather loosely refer to as a 'network' represents in an exemplary way the fundamental design dilemma for such systems: namely the tradeoff between the benefit of collaboration by the nodes (or agents) vs. the cost of (or constraints for) such collaboration. Security, authentication and trust are fundamental for the creation, maintenance and operation of any network. Yet, if their enforcement and monitoring come to an extreme, the benefits of the networked paradigm decrease and may be eliminated all together. This is the main reason for selecting the topic of security and trust in autonomic networks for this paper. As we recently emphasized in [7], security and trust are absolutely critical for the many wonderful and ubiquitous applications and systems that wireless network technologies promise. Simply put 'failure to address them will "kill" markets and the current momentum' [7]. Thus, on one hand the results and methods described here are important and innovative in their own right (i.e. for network security, information assurance and trust for wireless autonomic networks), but even more importantly they describe a circle of ideas and methods that have much wider applicability for autonomous networks.

Network security addresses such problems as detection of misbehaving agents, detection and classification of attacks by malicious agents, development of defensive and restorative strategies against attacks, development of network protocols and algorithms resilient to attacks [4,50,73,76]. From a systems and control perspective these problems are closely related to failure detection and classification, design for robust performance, and dynamic games between the attackers and the controllers (defenders) of the networked system. The models and performance requirements are substantially more complex than traditional systems control. Recently [40], network security has been correctly identified as part of network control and management. Thus, network security has been recognized as a managed quality metric, and this brings the entire subject much closer to dynamic, measurement based, feedback control.

Trust is important and critical for network security. It integrates with several components of network management, such as risk management, access control and authentication. Trust management is to collect, analyze and present trust-related evidence and to make assessments and decisions regarding trust relationships between entities in a network [13]. By analyzing, trust evaluation rules, such as local voting, we study how the 'trustworthiness' of the whole network evolves with time. Trust evaluation is identified as an iterated dynamic process (deterministic or stochastic); indeed a dynamic system on a time varying graph. The convergence of this process, its steady state and the speed of convergence are investigated. The dynamics and iterations involved in some simple cases are surprisingly similar to certain iterations that have created substantial excitement in the systems and controls community recently [2,45,64], in relation to collaborative robotics, swarms and formation control. But in addition here they involve substantial new extensions, in an exciting mixture of algebraic, statistical and dynamical systems methods and optimization. Some of our results are surprising, such as the discovery of phase transition phenomena, and the close relationship to recent advances in the statistical physics of spin-glasses [63].

Trust is interpreted as a set of relations among entities participating in network activities [9]. Trust establishment and maintenance in distributed and resource-constrained networks, such as mobile ad hoc networks (MANETs) [18,42], sensor networks and ubiquitous computing systems, is considerably more difficult [9] than in traditional hierarchical architectures, such as the Internet and wireless LANs centered on base stations and access points. In traditional networks, such as Internet, sources of trust evidence are centralized control servers, such as trusted third parties (TTPs) and authentication servers (ASs). Those servers are trusted and available all the time. In contrast, autonomic networks [25] have neither fixed infrastructures, nor centralized control servers. In these networks, the sources of trust evidence are peers, i.e. the entities that form the network. To manage trust in a distributed way has several advantages. Because of locality, it saves network resources (power, bandwidth, computation, etc.). It avoids the single point of failure problem as well.

Even in the conventional Internet, people have gradually realized the importance of distributed trust management as more and more people rely on network resources. Through the Internet, individuals can make their personal thoughts, reactions, and opinions easily accessible to the global community of Internet users. 'Word-of-mouth' is being given new significance by this unique property of the Internet [27]. One prominent example is *eBay*. The control of such distributed trust systems is much more difficult, as it includes trust evidence collection, policy specifications, evaluation rules, etc.

This paper is organized as follows. In Section 2 we describe our results on intrusion detection for worms and DDoS. In Section 3 we provide recent results on robust detection of misbehavior in wireless network protocols. Section 4, describes very recent results on the evaluation of IDS, and resolves the base-rate fallacy. Section 5 provides analysis of distributed trust dynamics, convergence and topology effects. In Section 6, we describe how trust can be used to induce cooperation between nodes and we introduce constrained coalitional games. Section 7, treats trust computation as a generalized shortest path problem in an ordered semiring. Finally, Section 8 provides very recent results on cooperative games and reputation formation in networks. Throughout the paper we point out promising research directions.

2. Intrusion Detection of Worms and DDoS

Intrusion detection systems (IDS) usually monitor and detect the misuse of network resources by keeping a series of statistics related to the normal or acceptable use of the network. These network statistics are computed from network data, collected by 'security sensors', which are specialized software or hardware-software systems that 'sniff' the network for specific data gathering (e.g. flow, usability, program system calls, etc.). Continuous monitoring of the network statistics is performed and as soon as the monitored statistics cross certain thresholds or violate a fixed policy on network usage an alarm is raised. Alarms can be raised by individual sensors or collaboratively by groups of sensors.

Sequential detection theory provides an ideal framework to analyze and propose new algorithms for the quickest change detection in the monitored statistics. In our work [8,21] we have used this approach to quickly detect attacks such as Spreading Worms and Distributed Denial of Service (DDoS). Due to the large scale of these attacks a distributed formulation, where sensors are placed in different parts of the network, is considered. We have also considered monitoring the hop count distribution for distance vector routing algorithms as an approach to detect attacks to the routing protocols of wireless ad hoc networks [21]. Change detection theory [12] was originally developed for failure detection in systems and control, for multi-model tracking systems, for network communication and control problems.

2.1. Change Detection for Worms

For clarity of presentation we will consider active worms as opposed to email worms. Active worms are programs that self-propagate across a network by exploiting vulnerabilities in widely used services offered by computers in the network. In order to locate the vulnerable computers, the worm probes different computer addresses at the specific port number of the service it is looking for. By exploiting the security flaw in the service offered by the computer, the worm can execute arbitrary code with elevated privileges, allowing it to copy and execute itself in the compromised machine. In order to reproduce, the worm scans for new vulnerable machines from each new compromised computer. The prevalence of active worms can be seen from some examples in recent years: Code Red I (July 2001), Code Red II (August 2001), NIMDA (September 2001), Sapphire, also known as Slammer (January 2003) and Blaster (August 2003).

The top three categories of computer attacks are directly related to worms and other self-propagating hybrid threats, which exploit multiple vulnerabilities across desktops and servers. An important requirement is to detect a worm as soon as possible in order to minimize the number of compromised hosts. A case example is the quick discovery and prompt action by System Administrators which prohibited Slapper from spreading further and prevented damage [44]. Some highly contagious worms can also have side effects such as BGP routing instabilities [26] when they reach their peak. Currently however, detection relies mostly on informal email discussion through a few key mailing lists. This process takes hours at a minimum, which is too slow for rapidly propagating worms. Furthermore [77] the spread of the theoretical flash or Warhol worms will be so fast that no human-driven communication will suffice for adequate identification of an outbreak before nearly complete infection is achieved. It is therefore critical to develop automated mechanisms for detecting worms based on their traffic patterns or other 'signatures'.

The self propagating code will try to use specific vulnerabilities that can be identified with certain port numbers. So in the rest of this section we assume that the traffic monitoring variable X is the connection attempts (probes) to a given TCP/UDP port number(s). We also assume most of the times a parametric pdf model $f(X)$ of the traffic observations. We have explored [8,21] the effect of aggregation from distributed sensors, motivated by the current infrastructure of distributed Intrusion Detection Systems [77].

2.1.1. Distributed Detection of A Change in the Mean

Clearly the simplest approach to change detection is to detect a change in the mean. Despite the abundance of techniques addressing the change detection problem, optimum schemes can mostly be found for the case where the data are independent and identically distributed (i.i.d.) and the distributions are completely known before and after the change time k_0 [61]. The cumulative sum (CUSUM) and the Shiryaev-Roberts statistics are the two most commonly used algorithms for change detection problems; we have applied both to this problem [8,21]. Let $\{X_k\}$ be the aggregate traffic from all the sensors in the network. To detect a change in the mean we assume $\{X_k\}$ is i.i.d with pdf $f^{(0)}$ before and $f^{(1)}$ after the change, such that the historical mean $E[f^{(0)}(X)]$ is less than the change mean $E[f^{(1)}(X)]$. For further details we refer to [8,21].

2.1.2. Detection of an Exponential Signal in Noise

Clearly detecting a change in the mean might give rise to several false alarms as there might be cases where the observed traffic increases during the normal operation of the network. Furthermore, the i.i.d assumption of the observations after the change is too strong because each infected host will try in general to scan the same number of hosts in a given interval of time, and as more and more hosts become infected X_k will increase with k. In particular we know from simple population dynamic models that a worm scanning uniformly at random the network will follow a logistic growth [77].

Let η be the population of infected hosts. Let r be the intrinsic growth rate (the growth rate when η is small) and let a be a given positive constant. Then the logistic growth satisfies the nonlinear ordinary differential equation

$$\frac{d\eta}{dt} = (r - a\eta)\eta \tag{1}$$

with solution

$$\eta(t) = \frac{N_0 B}{N_0 + (B - N_0)\mathrm{e}^{-rt}} \tag{2}$$

where $B = r/a$ and N_0 is the population at time 0. Since we are interested in detecting a worm as soon as possible we will be interested in the behavior of $\eta(t)$ when t is small, i.e. we consider the exponential growth

$$\eta(t) = N_0\mathrm{e}^{rt} \tag{3}$$

The equivalent discrete time recursion is

$$\eta(k\Delta t) := \eta_k^d = N_0 m^k \tag{4}$$

(d stands for 'discrete') where m is the discretized growth rate when η_d is small ($m = e^r$) and N_0 is the number of hosts compromised at $k = 0$.

For the detection problem we will assume that the values of N_0 and r (or m) are unknown. We will also consider a dummy signal η_k^{dummy} to represent any other growth pattern we want to discriminate (e.g. linear growth, a step function, etc.) from the growth of the worm η_k^d.

Let X_k denote the aggregate observation from all sensors at time k, i.e.

$$X_k = \sum_{l=1}^{L} X_{l,k} \tag{5}$$

We assume that the normal traffic aggregate is distributed as $f^{(0)}(x_1, \ldots, x_k)$.

Our main assumption is that the number of probes seen at the sensors will be proportional to the number of infected hosts $\alpha\eta_k^d$. The usual change detection hypothesis testing problem for the aggregate traffic (Eq. (5)) is:

$$H_0 : x_k = \eta_k^{\text{dummy}} + w_k \quad \text{when } 1 \leq k \leq M$$

$$H_1 : \begin{cases} x_k &= \eta_k^{\text{dummy}} + w_k \quad \text{when } 1 \leq k < k_0 \\ x_k &= \alpha\eta_k^d + w_k \quad \text{when } k_0 \leq k \leq M \end{cases}$$

However, we want k to restart at 1 whenever H_0 is accepted, so we use a sequential hypothesis test where the change time k_0 is implicitly given by the time at which the sequential test restarted and H_1 was accepted.

$$H_0 : x_k = \eta_k^{\text{dummy}} + w_k \quad \text{when } 1 \leq k \leq M$$
$$H_1 : x_k = \alpha\eta_k^d + w_k \quad \text{when } 1 \leq k \leq M$$

(a)Exponential signal detection in noise: Since we assume we do not know the parameters α, N_0 and m, we compute the generalized likelihood ratio (GLR) in a given time window $[1, \ldots, M]$ and compare it to a threshold h. We also assume the dummy signal has some unknown parameter β (e.g. the slope in a linear growth). Therefore detection of the signal $\alpha\eta_k^d$ in noise w_k is achieved with the test:

$$\frac{\sup_{\alpha, N_0, m} f^{(0)}(x_k - \alpha\eta_k^d)}{\sup_{\beta} f^{(0)}(x_k - \eta_k^{\text{dummy}})} \underset{H_0}{\overset{H_1}{\gtrless}} h \tag{6}$$

(b)Nonparametric regression detection: So far we have always been assuming a parametric distribution $f^{(0)}(x_1, \ldots, x_k)$ for the normal traffic. This assumption is valid for a wide number of ports as the traffic seen can be regular. However in some cases the real distribution can be quite difficult to obtain. For example the number of probes seen to port 80 (WWW) or port 21 (FTP) for computers providing those services can exhibit long range dependence and multifractal behavior, difficult to capture with a parametric model. To deal with some of the more complicated traffic observations we have used [8,21] a heuristic non-parametric change detection algorithm similar to the problem of detection of an exponential signal in noise.

2.2. Change Detection for DDoS Attacks

Almost all DDoS attacks involve multiple networks and attack sources, many of which have spoofed IP addresses to make detection even harder. An attempt of the victim to choke off the offending traffic requires network administrators to call upstream service providers, alerting them of the attack and having them shut down the traffic. That process has to be repeated all the way back to every attack source. So although DDoS are easily identified at the victim's site, it is natural to extend the quickest detection problem to transit networks (ISPs) for faster response to an attack.

At the ISP level, traffic anomalies are difficult to detect in the aggregated network traffic. Examination at per-flow basis at the IP level cannot usually scale up to the high-speed links in the transit networks. Thus we are interested only in passively monitoring the aggregate traffic, without the need to store header information from the packets transmitted through the network.

We have introduced [8,21] a new approach for identifying Distributed Denial of Service attacks by a set of nodes in a transit network. The basic idea is that at each highly connected node the data tends to aggregate from the distributed sources toward the destination, giving a sense of directionality to the attack. This directionality idea provides a framework to design change detection algorithms that are less sensitive to changes in the average intensity of the overall traffic and focus on differentiating random fluctuations of the network traffic versus fluctuations where there is a clear change in the direction of the flow at a given node. We are considering packets in a very broad and general way, but clearly our approach can be extended to monitor certain specific packet types given the protocol; for example measuring only TCP SYN-ACK response packets for identifying a reflected DDoS attack, or ICMP packets for identifying ping floods.

Assume we are monitoring node d in Fig. 1. Let $X_k^{d,m}$ denote the stochastic process representing the total number of packets sent by d through the link (d,m) (an ordered pair) at time step k, where $m \in \mathcal{N}(d)$

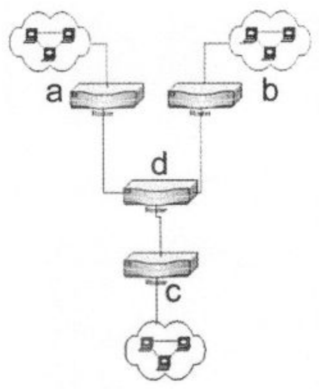

Fig. 1. A transit network composed of nodes a, b, c and d.

denotes a neighbor of d, and $\mathcal{N}(d)$ the set of neighbors of d. Let X_k^d denote the vector with the elements $X_k^{d,m}$ and let

$$\theta_0^d := \begin{bmatrix} E_0[X_k^{d,a}] \\ E_0[X_k^{d,b}] \\ E_0[X_k^{d,c}] \end{bmatrix} \tag{7}$$

We will be interested in changes of the form:

$$\theta_0^d + \nu \Upsilon_m \tag{8}$$

where ν is a non-negative scalar and Υ_m is one of the standard basis vectors. So in Fig. 1, if node d suddenly starts a broadcast, there will be a change in the mean of all processes; we are not interested in such a change. Instead, if there are attackers in the subnetworks attached to b and c, and they target a host in the network attached to a by flooding it, there will be a change in the direction Υ_a. Testing directions should help us in discriminating unwanted false alarms.

To formalize our ideas we consider the framework discussed in [12] of change detection in a known direction but unknown magnitude of the change. Our problem is a little bit different in that we are considering an M-ary sequential hypothesis testing problem and in that we do not allow changes with negative or zero values for ν, i.e. we impose the restriction $\nu \geq 0$.

Thus the resulting change detection problem is:

$$\theta^d(k) = \begin{cases} \theta_0^d & \text{when} \quad k < t_{\text{change}} \\ \theta_0^d + \nu\Upsilon_a & \text{or} \\ \theta_0^d + \nu\Upsilon_b & \text{or} \\ \theta_0^d + \nu\Upsilon_c & \text{when} \quad k \geq t_{\text{change}} \end{cases} \tag{9}$$

where t_{change} is an unknown time step when the change occurs.

Since we have an unknown parameter ν we follow the generalized likelihood ratio (GLR) for a multi-hypothesis test: a test for each possible direction Υ_m, vs. the null hypothesis: a change in all directions Υ_d. The null hypothesis is selected for discriminating a change in one direction vs. a change of the overall traffic of the network either as an increase or decrease:

$$g_k^{d,m} = \max_{1 \leq j \leq k} \log \frac{\sup_{\nu \geq c_1} \Pi_{i=j}^k f_{\theta_0^d + \nu\Upsilon_m}(X_i^d)}{\sup_\lambda \Pi_{i=j}^k f_{\theta_0^d + \lambda\Upsilon_d}(X_i^d)}$$

where λ is a scalar not necessarily greater than a positive constant c_1 unlike ν (i.e. we allow also for a decrease in the overall network traffic). The threshold $h^{d,m}$ for each of the tests is selected given a fixed false alarm rate probability.

To stop the test we can run all hypothesis in parallel and only the test $g_k^{d,m}$ that reaches its given threshold is stopped. However this is a heuristic procedure as optimal solutions to the problem of sequential testing of more than two hypotheses are, in general, intractable. A more elaborate stopping rule is presented in [30] with a proof of asymptotic optimality as the decision risks (or error probabilities) go to zero. For further details we refer to [8,21].

2.3. Sensor Fusion

So far we have been focusing on detecting a change in a single node. One of the main advantages in having several nodes under monitoring is that we can perform an aggregation of the statistics between the different nodes in order to decrease detection delay given a fixed false alarm rate probability. In particular if we are monitoring nodes far away from the destination, most of the local statistics will not yield an alarm and the attack might be unnoticed. The alarm aggregation can be performed by several methods. Here we propose a simple heuristic that will apply to any distance vector routing protocol.

We want a mechanism to aggregate the different statistics at each monitored node, taking into account that the computed statistics for all nodes can vary to different scales of magnitude yielding a biased addition. To cope with this problem we compute the normalized statistic $\varphi_k^{d,m} := \frac{g_k^{d,m}}{h^{d,m}}$. If none of our monitored nodes has raised an alarm, the number of monitored nodes will be bounded by $\sum_d \varphi_k^{d,m}$. This can be in turn interpreted as a new upper bound for a collective threshold which can be selected given a false alarm rate probability.

Selecting which statistics to add is the key issue. In keeping with our directionality framework we combine only the statistics relating two or more nodes to a common destination.

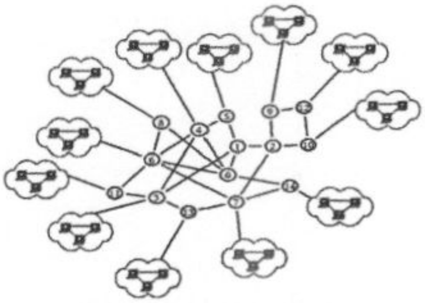

Fig. 2. The transit network.

We now apply this formulation to the case of two monitored nodes (a natural extension follows for several nodes). Suppose we monitor nodes 6 and 3 in the transit network model shown in Fig. 2, where the transit network consists of 15 routers numbered from 0 to 14. Each cloud represents a subnetwork with its own routing domain.

The routing tables required for the aggregation algorithm are given in Tables 1 and 2. By simple inspection of the routing tables we see that we need to correlate the link (6,0) with (3,1) because nodes 6 and 3 use them (respectively) to reach nodes 0, 1 and 14. Similarly, the link (6,11) must be correlated with (3,11), link (6,4) with (3,4), link (6,7) with (3,13), (6,7) with (3,1) and (6,8) with (3,11).

With this approach not only we can improve our chances to detect "buried" attacks in single links by correlating statistics, but also diminish the impact of false alarms originating from individual nodes. See [8,21] for details.

Table 1. Routing table for node 6

Link	Routing to nodes
(6,7)	7,13,2,10,12,9
(6,0)	0,14,1
(6,4)	4,3,5
6,11)	11,3
(6,8)	8
(6,subnetwork)	

Table 2. Routing table for node 3

Link	Routing to nodes
(3,1)	1,0,2,14,10,9,12
(3,13)	7,13
(3,4)	4,5
(3,11)	11,6,8
(3,subnetwork)	

3. Robust Detection of Misbehavior

The problem of deviation from legitimate protocol operation in wireless networks and the efficient detection of such behavior has become a significant issue in recent years [4,18,42,50,73,76]. In our work we have addressed and quantified the impact of MAC layer attacks [23,24,68–70], routing layer attacks [93], that aim at disrupting critical network functionalities and information flow in autonomous wireless networks. We describe here our work on MAC misbehavior detection.

In the distributed coordinating function (DCF) of the IEEE 802.11 MAC protocol, coordination of channel access for contending nodes is achieved via carrier sense multiple access with collision avoidance (CSMA/CA). A node with a packet to transmit selects a random back-off value b uniformly from the set $\{0, 1, \ldots, W-1\}$, where W is the (fixed) size of the contention window. The back-off counter decreases by one at each time slot that is sensed to be idle and the node transmits after b idle slots. In case the channel is perceived to be busy in one slot, the back-off counter stops momentarily. After the back-off counter is decreased to zero, the transmitter can reserve the channel for the duration of data transfer. First, it sends a request-to-send (RTS) packet to the receiver, which responds with a clear-to-send (CTS) packet. Thus, the channel is reserved for the transmission. Both RTS and CTS messages contain the intended duration of data transmission in the duration field. Other hosts overhearing either the RTS or the CTS are required to adjust their network allocation vector (NAV) that indicates the duration for which they will defer transmission. This duration includes the SIFS intervals, data packets and acknowledgment frame following the transmitted data frame. An unsuccessful transmission instance due to collision or interference is denoted by lack of CTS or ACK for the data sent and causes the value of contention window to double. If the transmission is successful, the host resets its contention window to the minimum value W.

IEEE 802.11 DCF favors the node that selects the smallest back-off value among a set of contending nodes. Therefore, a malicious or selfish node may choose not to comply to protocol rules by selecting small back-off intervals, thereby gaining significant advantage in channel sharing over regularly behaving, honest nodes. Moreover, due to the exponential increase of the contention window after each unsuccessful transmission, non-malicious nodes are forced to select their future back-offs from larger intervals after every access failure. Therefore the chance of their accessing the channel becomes even smaller.

MAC layer protocol misbehavior has been studied in various scenarios and mathematical frameworks. The random nature of access protocols coupled with the highly volatile nature of the wireless medium poses the major obstacle in developing a unified framework for misbehavior detection. The goals of a misbehaving peer can range from exploitation of available network resources for its own benefit up to network disruption. An efficient Intrusion Detection System should detect a wide range of misbehavior policies with an acceptable False Alarm rate; a major challenge.

The current literature offers two major approaches in the field of misbehavior detection. The first provides solutions based on modification of the current IEEE 802.11 MAC layer protocol by making each protocol participant aware of the backoff values of its neighbors. A different line of thought is followed in [69–71], where the authors propose a misbehavior detection scheme without making any changes to the actual protocol. The authors in [69,70] address the detection of an adaptive intelligent attacker by casting the problem of misbehavior detection within the minimax robust detection framework. System performance is measured in terms of number of required observation samples to derive a decision (detection delay).

It is important to note that the parameters used for deriving a decision of whether a protocol participant misbehaves or not should be carefully chosen. For example, choosing the percentage of time the node accesses the channel as a misbehavior metric can result in a high number of false alarms due to the fact that the other protocol participants might not have anything to transmit within a given observation period. This could easily lead to false accusations of legitimate nodes that have large amounts of data to send.

In our work we have derived analytical performance bounds of two proposed schemes for detecting random access misbehavior: DOMINO [71] and SPRT-based tests [23,24,68–70] and have shown the optimality of SPRT against the worst-case adversary for all configurations of DOMINO. Following the main idea of DOMINO, we have introduced [24] a nonparametric CUSUM statistic that shares the same intuition as DOMINO but gives better performance for all configurations of DOMINO.

3.1. Sequential Detection

Consider monitoring the behavior of node A for the single-hop communication with node B. We assume that any node within the transmission range of A or B observes the same sequence of measurements of back-off values used by A. Since the sequence of observations is the same, the procedure that will be described in the sequel can take place in any of these observer nodes. Since the back-off measurements are enhanced by an additional sample each time A attempts to access the channel, an on-line sequential scheme is natural. The basis of such a scheme is a sequential detection test that is implemented at an observer node. The objective of the detection test is to derive a decision as to whether or not a misbehavior occurs as fast as possible, namely with the least possible number of observation samples.

The probability of false alarm and the probability of missed detection constitute inherent tradeoffs in such a scheme. Clearly, we can obtain small values for both by accumulating more information, that is, at the expense of larger detection delay. A logical compromise would be to prescribe some maximal allowable values for the two error probabilities and attempt to *minimize* the expected detection delay. Expressing this problem under a more formal setting, we are interested in finding a sequential test $\mathcal{D} = (N, d_N)$ that solves the following constrained optimization problem: $\inf_{N, d_N} \mathbb{E}_1[N]$ under the constraints $\mathbb{P}_0[d_N = 1] \leq \alpha$; $\mathbb{P}_1[d_N = 0] \leq \beta$, where $\mathbb{P}_i, \mathbb{E}_i$ denote probability and expectation under hypothesis $\mathbf{H}_i, i = 0, 1$, and $0 < \alpha, \beta < 1$ are the prescribed values for the probability of false alarm and miss detection respectively.

This setup was first proposed in [88] where the Sequential Probability Ratio Test (SPRT) was introduced for its solution. Optimality of SPRT is assured *only* when the data are i.i.d. under both hypotheses [89]. In order to use the SPRT test it is necessary to specify both probability density functions $f_i(x), i = 0, 1$ under the two hypotheses. Although the pdf $f_0(x)$ of a legitimate node is known, this is not the case for an attacker. Furthermore, specifying a candidate density $f_1(x)$ for an attacker without some proper analysis may result in serious performance degradation if the attacker's strategy diverges from our selection.

In order to be able to propose a specific detection rule we need to clarify and mathematically formulate the notion of an 'attack'. We should place our main emphasis to attacks that incur large gains for the attacker (result in higher chances of channel access). Besides, if we assume that the detection of an attack is followed by communication of the attack event further in the network so as to launch a network response, it would be rather inefficient for the algorithm to consider less significant attacks and initiate responses for them. Instead, it is meaningful for the detection system to focus on encountering the most significant attacks and at the same time not to consume resources of any kind (processor power, energy, time or bandwidth) for dealing with attacks whose effect on performance is rather marginal.

3.2. Minimax Robust Detection: The Uncertainty Class

We need to cope with the encountered (statistically) uncertain operational environment of a wireless network. Hence, it is desirable to rely on robust detection rules that would perform well regardless of these uncertainties. In our work [24,68–70], we have adopted a minimax robust detection approach, where the goal is to optimize performance for the worst-case instance of uncertainty. We identify the least favorable operating point of the system in the presence of uncertainty and subsequently find the strategy that optimizes system performance when operating at that point. In our case, the least favorable operating point corresponds to the worst-case instance of an attack and the optimal strategy amounts to the optimal detection rule. System performance is measured in terms of the number of required observation samples.

Implicit in the minimax approach is the assumption that the attacker has full knowledge of the employed detection rule. Thus, it can create a misbehavior strategy that maximizes the number of required samples for misbehavior detection. Our approach addresses the case of an intelligent attacker that can adapt its policy to avoid detection.

According to the IEEE 802.11 MAC standard, the back-off for each legitimate node is selected from a set of values in a contention window interval based on a uniform distribution. The length of the contention window is 2^iW for the ith retransmission attempt, where W is the minimum contention window. In general, some back-off values will be selected uniformly from $[0, W]$ and others will be selected uniformly from intervals $[0, 2^iW]$, for $i = 1, \ldots, I_{\max}$ where $I_{\max}$ is the maximum number of re-transmission attempts. Without loss of generality, we can scale down a back-off value that is selected uniformly in $[0, 2^iW]$ by a factor of 2^i, so that all back-offs can be considered to be uniformly selected from $[0, W]$. An attack strategy is mapped to a probability density function, which the attacker uses to select his back-off value. We consider continuously back-logged nodes that always have packets to send. Thus, the gain of the attacker is signified by the percentage of time in which it obtains access to the medium. This in turn depends directly on the relative values of back-offs used by the attacker and by the legitimate nodes.

Let us first compute the probability P_1 of the attacker to access the channel as a function of the pdfs f_1 and f_0. Let us observe the backoff times during a fixed period T that *does not include* transmission intervals. Consider first the case of one misbehaving and one legitimate node and assume that within the time period T, we observe $X_1, \ldots, X_N$, N samples of the attacker's backoff and $Y_1, \ldots, Y_M$, M samples of the legitimate node's backoffs. It is then clear that the attacker's percentage of accessing the channel during the period T is $N/(N+M)$. In order to obtain the desired probability we simply need to compute the limit of this ratio as $T \to \infty$. We use the fact that

$$X_1 + \cdots + X_N \le T < X_1 + \cdots + X_{N+1}$$
$$Y_1 + \cdots + Y_M \le T < Y_1 + \cdots + Y_{M+1},$$

and let $T \to \infty$ (resulting in $N, M \to \infty$). Finally, by applying the Law of Large Numbers, we conclude that

$$\mathrm{P}_1 = \lim_{N,M\to\infty} \frac{N}{N+M} = \frac{\frac{1}{\mathbb{E}_1[X]}}{\frac{1}{\mathbb{E}_1[X]} + \frac{1}{\mathbb{E}_0[Y]}} \tag{10}$$

Using exactly similar reasoning the probability P_1, for one misbehaving node against n legitimate ones, is

$$\mathrm{P}_1 = \frac{\frac{1}{\mathbb{E}_1[X]}}{\frac{1}{\mathbb{E}_1[X]} + \frac{n}{\mathbb{E}_0[Y]}} = \frac{1}{1 + n\frac{\mathbb{E}_1[X]}{\mathbb{E}_0[Y]}} = \frac{1}{1 + n\frac{2\mathbb{E}_1[X]}{W}}. \tag{11}$$

If the attacker were legitimate then $\mathbb{E}_1[X] = \mathbb{E}_0[Y]$ and his probability of accessing the channel, from Eq. (11), would have been $1/(n+1)$. It is therefore clear that whenever

$$\mathbb{E}_1[X] = \varepsilon\mathbb{E}_0[Y], \quad \text{with } \varepsilon \in (0,1) \tag{12}$$

the attacker enjoys a gain compared to legitimate nodes since

$$\mathrm{P}_1 = \eta\frac{1}{n+1} > \frac{1}{n+1}, \eta = \frac{1+n}{1+\varepsilon n} \in (1, n+1). \tag{13}$$

His probability of accessing the channel is greater than the probability of any legitimate node by a factor $\eta > 1$.

Using this simple model we are now able to quantify the notion of an 'attack'. Let η be a quantity that satisfies $1 < \eta < n+1$ and consider the class $\mathcal{F}_\eta$ of all pdfs that induce a probability P_1 of accessing the channel that is no less than $\eta/(n+1)$. From (12), (13), the class $\mathcal{F}_\eta$ is explicitly defined as

$$\mathcal{F}_\eta = \left\{ f_1(x) : \int_0^W x f_1(x)\mathrm{d}x \le \frac{1 - \frac{\eta}{n+1}}{n\frac{\eta}{n+1}} \frac{W}{2} \right\} \tag{14}$$

This class includes all possible attacks for which the incurred relative gain exceeds the legitimate one by $(\eta - 1) \times 100\%$. The class $\mathcal{F}_\eta$ is the *uncertainty class* of the robust approach and η is a tunable parameter. We define the severity of the attack by changing the gain η. Values of η larger but close to 1 are equivalent to low-impact attacks whereas values significantly larger than 1 are equivalent to DoS attacks.

3.3. Minimax Robust Detection: The Worst-Case Attack

Hypothesis $\mathbf{H}_0$ concerns legitimate operation and thus the corresponding *pdf* $f_0(x)$ is the uniform one. Hypothesis $\mathbf{H}_1$ corresponds to misbehavior with unknown *pdf* $f_1(x) \in \mathcal{F}_\eta$. The performance of the detection scheme is quantified by the average number of samples $\mathbb{E}_1[N]$ needed until a decision is reached, which is clearly a function of the adopted detection rule $\mathcal{D}$ and the *pdf* $f_1(x)$: $\mathbb{E}_1[N] = \varphi(\mathcal{D}, f_1)$.

Let $\mathcal{T}_{\alpha,\beta}$ denote the class of all sequential tests for which the false alarm and missed detection probabilities do not exceed some specified levels α and β respectively. In the context of minimax robust detection, the goal is to optimize performance in the presence of worst-case attack, that is, solve the min-max problem

$$\inf_{\mathcal{D}\in\mathcal{T}_{\alpha,\beta}} \sup_{f_1\in\mathcal{F}_\eta} \varphi(\mathcal{D}, f_1). \tag{15}$$

A saddle point $(\mathcal{D}^*, f_1^*)$ of φ consists of a detection scheme $\mathcal{D}^*$ and an attack distribution f_1^*. It is specified in

Theorem 1. Let the gain factor $\eta \in (1, n+1)$ and the maximal allowable decision error probabilities α, β be given. Then the pair $(\mathcal{D}^*, f_1^*)$ which *asymptotically* (for small values of α, β) solves the saddle point problem is

$$f_1^*(x) = \frac{\mu}{W} \frac{\mathrm{e}^{\mu}(1 - \frac{x}{W})}{\mathrm{e}^{\mu} - 1}, \tag{16}$$

where $\mu > 0$ is the solution to the following equation

$$2\left(\frac{1}{\mu} - \frac{1}{\mathrm{e}^{\mu} - 1}\right) = \frac{1 - \frac{\eta}{n+1}}{n\frac{\eta}{n+1}}. \tag{17}$$

The corresponding decision rule $\mathcal{D}^* = (N^*, d_{N^*})$ is the SPRT test that discriminates between $f_1^*(x)$ and $f_0(x)$.

For the complete proof, we refer the reader to [70]. Our robust detection approach captures the case of an intelligent attacker. The SPRT algorithm is part of the intrusion detection system module that resides at an observer node.

3.4. Colluding Attackers

Applying the same min-max robust approach to the case of two colluding (cooperating) attackers, the following expression is derived for the access policy of the collaborating attackers:

$$f_{12}^*(x_1, x_2) = \mathrm{e}^{-1-\lambda}\mathrm{e}^{-\mu \min(x_1, x_2)/W} \tag{18}$$

where x_1 and x_2 represent the backoff values of malicious nodes 1 and 2. For further details on performance of colluding attackers we refer to [67,69].

3.5. Experimental Results

The backoff distribution of an optimal attacker was implemented in the network simulator Opnet and tests were performed for various levels of false alarms. The experimental results presented here correspond to the scenario consisting of two legitimate and one selfish node competing for channel access. It is important to mention that the resulting performance comparison of DOMINO, CUSUM and SPRT does not change for any number of competing nodes, SPRT always exhibits the best performance. In order to demonstrate the performance of all detection schemes for more aggressive attacks, we present the results for the scenario where the attacker attempts to access the channel for 60% of the time (as opposed to 33% if it was behaving legitimately). The results for both legitimate and malicious behavior were collected over a fixed period of 100s. In order to obtain fair performance comparison, a performance metric different from the one in [70] was adopted. The evaluation was performed as a tradeoff between the average time to detection and the average time to false alarm.

We evaluated the performance of the SPRT using the same parameters as in the theoretical analysis. DOMINO was evaluated for fixed $\gamma = 0.9$, which corresponds to the value used in the experimental evaluation in [71]. In order to compare the performance to SPRT, we varied the value of K (determines the number of false alarms). We computed the performance of DOMINO for 2 different values of the parameter m. As it can be seen from Fig. 3, SPRT outperforms DOMINO for all values of K and m. We note that the best performance of DOMINO was obtained for $m = 1$ (the detection delay is smaller when the decision is made after every sample). Therefore, we adopted $m = 1$ for further analysis of

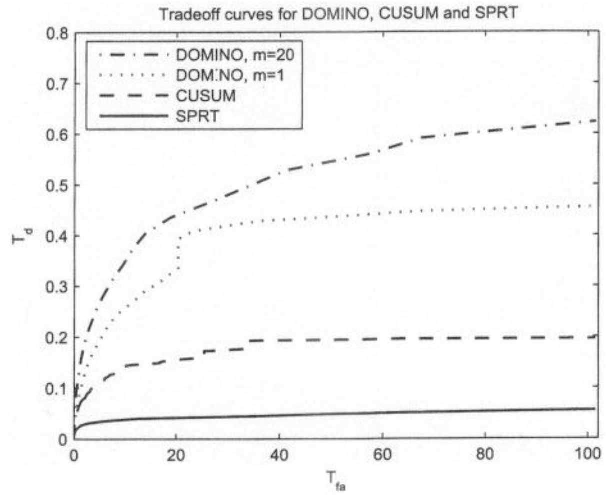

Fig. 3. Tradeoff curves for each of the proposed algorithms. DOMINO has parameters $\gamma = 0.9$ and $m = 1$ while K is the variable arameter. The nonparametric CUSUM algorithm has as variable parameter c and the SPRT has $b = 0.1$ and a is the variable parameter.

DOMINO. We found that the optimal value of γ is equal to 0.7 by experimental evaluation and we adopted this as the optimal parameter for DOMINO. With the optimal values of γ and m we performed final evaluations of DOMINO, CUSUM and SPRT, see Fig. 4. We observe that even for the optimal parameters of DOMINO, the SPRT outperforms it for all values of K.

We also compared the performance of collaborating attackers vs. a single attacker; results are shown in Fig. 5.

4. Evaluation of Intrusion Detection Systems

Consider a company that, in an effort to improve its information technology security infrastructure, wants to purchase either intrusion detector 1 ($\mathcal{IDS}_1$) or intrusion detector 2 ($\mathcal{IDS}_2$). Furthermore, suppose that the algorithms used by each IDS are kept private and therefore the only way to determine the performance of each IDS (unless some reverse engineering is done [54]) is through empirical tests determining how many intrusions are detected by each scheme while providing an acceptable level of false alarms. Suppose these tests show with high confidence that $\mathcal{IDS}_1$ detects one-tenth more attacks than $\mathcal{IDS}_2$ but at the cost of producing one hundred times more false alarms. The company needs to decide based on these estimates, which IDS will provide the best return on investment for their needs and environment.

This general problem is more concisely stated as the intrusion detection evaluation problem, and its solution usually depends on several factors. The most basic of these factors are the *false alarm rate* and the *detection rate*, and their tradeoff can be intuitively analyzed with the help of the *receiver operating characteristic* (ROC) curve [34,53,55,56,90]. However, as pointed out in [5,36,39], the information provided by the detection rate and the false alarm rate alone might not be enough to provide a good evaluation of the performance of an IDS. Therefore, the evaluation metrics need to consider the environment the IDS is going to operate in, such as the maintenance costs and the hostility of the operating environment (the likelihood of an attack). In an effort to provide such an evaluation method, several performance metrics such as the *Bayesian detection rate* [5], *expected cost* [36],

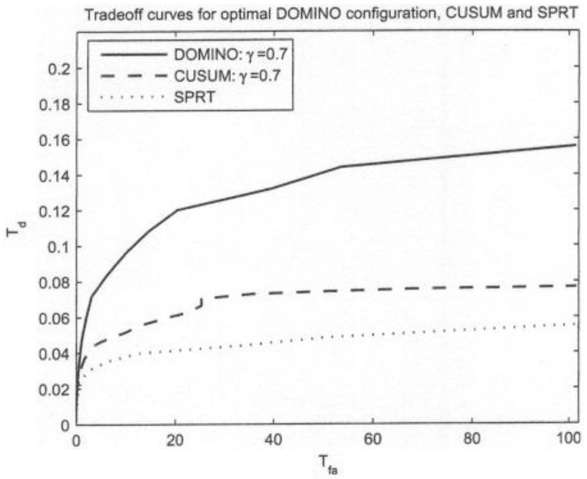

Fig. 4. Tradeoff curves for best performing DOMINO configuration with $\gamma = 0.7$, best performing CUSUM configuration with $\gamma = 0.7$ and SPRT.

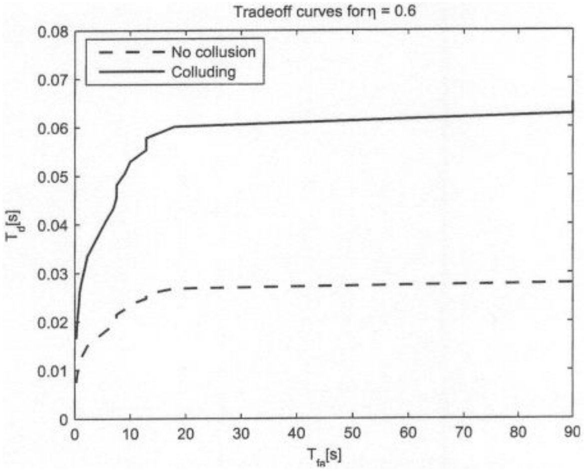

Fig. 5. Tradeoff curves for $\eta = 0.6$; detection times for colluding nodes are up to 2 times longer than for a single node with identical strategy.

sensitivity [28] and *intrusion detection capability* [39], have been proposed.

Yet despite the fact that each of these performance metrics makes their own contribution to the analysis of intrusion detection systems, they are rarely applied in the literature when proposing a new IDS. It is our belief that the lack of widespread adoption of these metrics stems from two main reasons. First, each metric is proposed in a different framework (e.g. information theory, decision theory, cryptography, etc.) and in a seemingly *ad hoc* manner. Therefore an objective comparison between the metrics is very difficult. The second reason is that the proposed metrics usually assume the knowledge of some uncertain parameters like the likelihood of an attack, or the costs of false alarms and missed detections. Moreover, these uncertain parameters can also change during the operation of an IDS. Therefore the evaluation of an IDS under some (wrongly) estimated parameters might not be of much value.

More importantly, there does not exist a security model for the evaluation of IDSs. Several researchers have pointed out the need to include the resistance against attacks as part of the evaluation of an IDS [41,52,66,74,79,80]. However, the traditional evaluation metrics are based on ideas developed for non-security related fields and they do not take into account the role of an adversary and the evaluation of the system against this adversary.

In our work, we have introduced a framework for the evaluation of IDSs in order to address these concerns. First, we identify the intrusion detection evaluation problem as a multi-criteria optimization problem. This framework lets us compare several of the previously proposed metrics in a unified manner. To this end, we recall that there are in general two ways to solve a multi-criteria optimization problem. The first approach is to combine the criteria to be optimized in a single optimization problem. We then show how the intrusion detection capability, the expected cost and the sensitivity metrics all fall into this category. The second approach is to evaluate a tradeoff curve. We show how the Bayesian rates and the ROC curve analysis are examples of this approach.

To address the uncertainty of the parameters assumed in each of the metrics, we have developed [22] a graphical approach that allows the comparison of the IDS metrics for a wide range of uncertain parameters. For the single optimization approach we showed in [22] how the concept of *isolines* can capture in a single value (the slope of the isoline) the uncertainties like the likelihood of an attack and the operational costs of the IDS. For the tradeoff curve approach, we introduced in [22] a new tradeoff curve we call the intrusion detector operating characteristic (IDOC). We believe the IDOC curve combines in a single graph all the relevant (and intuitive) parameters that affect the practical performance of an IDS.

We introduce here a robust evaluation approach in order to deal with the adversarial environment the IDS is deployed in. In particular, we do not want to find the best performing IDS on average, but the IDS that performs best against the worst type of attacks. To that end we extend our graphical approach to model the attacks against an IDS. In particular, we show how to find the best performing IDS against the worst type of attacks. This framework allows us to reason about the security of the IDS evaluation and the proposed metric against adaptive adversaries.

In [5] Axelsson pointed out that one of the causes for the large amount of false alarms that intrusion detectors generate is the enormous difference between the amount of normal events compared to the small amount of intrusion events. Intuitively, the *base-rate fallacy* states that because the likelihood of an attack is very small, even if an IDS fires an alarm, the likelihood of having an intrusion remains relatively small. Formally, when we compute the posterior probability of intrusion (a quantity known as the *Bayesian detection rate*, or the *positive predictive value* (PPV)) given that the IDS fired an alarm, we obtain:

$$\begin{aligned}\text{PPV} &\equiv \Pr[I=1|A=1]\\ &= \frac{\Pr[A=1|I=1]\Pr[I=1]}{\Pr[A=1|I=1]\Pr[I=1]+\Pr[A=1|I=0]\Pr[I=0]}\\ &= \frac{P_D p}{(P_D-P_{FA})p+P_{FA}}\end{aligned}$$

If the rate of incidence of an attack is very small, for example $p = 10^{-5}$, and if our detector has $P_D = 1$ and $P_{FA} = 0.01$, then $\Pr[I = 1|A = 1] = 0.000999$. That is on average, of 1000 alarms, only one would be a real intrusion.

It is easy to demonstrate that the PPV value is maximized when the false alarm rate goes to zero, even if the detection rate also does! Therefore [5] we require a trade-off between the PPV value and the *negative predictive value* (NPV):

$$\begin{aligned}\text{NPV} &\equiv \Pr[I=0|A=0]\\ &= \frac{(1-p)(1-P_{FA})}{p(1-P_D)+(1-p)(1-P_{FA})}\end{aligned}$$

4.1. Graphical Analysis

Our new graphical framework [22] allows the comparison of different metrics in the analysis and evaluation of IDSs. This graphical framework can be used to adaptively change the parameters of the IDS based on its actual performance during operation. The framework also allows for the comparison of different IDSs under different operating environments. Throughout this section we use one of the ROC curves analyzed in [36] and in [39]; mainly the ROC curve describing the performance of the Columbia team intrusion detector for the 1998 DARPA IDS evaluation [57].

4.1.1. *Visualizing the Expected Cost: The Minimization Approach*

The biggest drawback of the expected cost approach is that the assumptions and information about the likelihood of attacks and costs might not be known a priori. Moreover, these parameters can change dynamically during the system operation. It is thus desirable to be able to tune the uncertain IDS parameters based on feedback from its actual system performance in order to minimize $\mathbf{E}[C(I,A)]$.

We select the use of ROC curves as the basic 2-D graph because they illustrate the behavior of a classifier without regard to the uncertain parameters, such as the base-rate p and the operational costs $C(i,j)$. The ROC curve decouples the classification performance from these factors [65]. In the graphical framework, the relation of these uncertain factors with the ROC curve of an IDS will be reflected in the *isolines* of each metric, where isolines refer to lines that connect pairs of false alarm and detection rates such that any point on the line has equal expected cost. The evaluation of an IDS is therefore reduced to finding the point of the ROC curve that intercepts the optimal isoline of the metric.

Under the assumption of constant costs, we can see that the isolines for the expected cost $\mathbf{E}[C(I,A)]$ are in fact straight lines whose slope depends on the ratio between the costs and the likelihood ratio of an attack. Formally, if we want the pair of points (P_{FA1}, P_{D1}) and (P_{FA2}, P_{D2}) to have the same expected cost, they must be related by the equation:

$$m_{C,p} \equiv \frac{P_{D2} - P_{D1}}{P_{FA1} - P_{FA2}} = \frac{1-p}{p}\frac{C(0,1) - C(0,0)}{C(1,0) - C(1,1)} = \frac{1-p}{p}C$$

where C is the ratio between the costs, and $m_{C,p}$ is the slope of the isoline. The optimal operating point in the ROC is determined by the slope of the isolines, which in turn is determined by p and C. Therefore we can readily check how changes in the costs and in the likelihood of an attack will impact the optimal operating point.

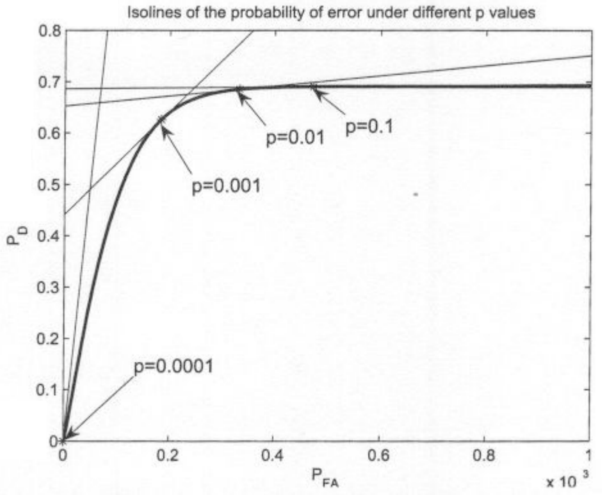

Fig. 6. As the base-rate p decreases, the slope of the optimal isoline increases.

4.1.2. *The Base-Rate Fallacy Implications on the Costs of an IDS*

In Fig. 6 we can see how as p decreases, the optimal operating point of the IDS tends to $(P_{FA}, P_D) = (0,0)$ (the evaluator must decide not to use the IDS for its current operating environment). Therefore, for small base-rates the operation of an IDS will be cost efficient only if we have an appropriate small C^* such that $m_{C^*,p^*} \leq m^c$. A small C^* results if the cost of a false alarm is much smaller than the cost of a missed detection: $C(1,0) \gg C(0,1)$.

4.1.3. *The Intrusion Detector Operating Characteristic: The Tradeoff Approach*

Although the graphical analysis introduced so far can be applied to analyze the cost efficiency of several metrics, the intuition for the tradeoff between the PPV and the NPV is still not clear. Therefore we have extended the graphical approach [19,20,22] by introducing a new pair of isolines, those of the PPV and the NPV metrics. It turns out that the most relevant metrics to use for a tradeoff in the performance of an IDS are PPV and P_D. However, even when we select for tradeoff PPV and P_D, the isoline analysis has still one deficiency when compared with the isoline performance analysis of the previous section – there is no efficient way to represent how the PPV changes with different p values. In order to solve this problem we introduced [19,20,22] the Intrusion Detector Operating Characteristic (IDOC) as a graph that shows how the two variables of interest: P_D and PPV are related under different base-rate values of interest. An example of an IDOC curve is presented in Fig 7.

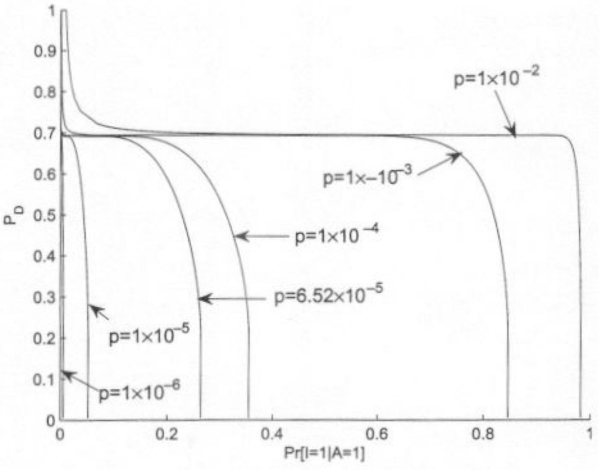

Fig. 7. IDOC example.

We believe that the IDOC provides a better way to evaluate IDS systems than most previously proposed metrics, because it compares tradeoffs that are easier to interpret.

4.2. Threat Models and Robust Evaluation of IDS

Traditional evaluation of intrusion detection schemes assumes that an intruder behaves similarly before and after the implementation of the IDS (i.e. a non-adaptive intruder). Now consider an intruder who adapts its attack when it faces a target system which hosts a given IDS.

For our evaluation analysis we assume three quantities that can be, up to a certain extent, controlled by the intruder. They are the base-rate p, the false alarm rate P_{FA} and the detection rate P_D. The base-rate can be modified by controlling the frequency of attacks. The perceived false alarm rate can be increased if the intruder finds a flaw in any of the signatures of an IDS that allows him to send maliciously crafted packets that trigger alarms at the IDS but that look benign to the IDS operator. Finally, the detection rate can be modified by the intruder with the creation of new attacks whose signatures do not match those of the IDS, or by evading the detection scheme, e.g. mimicry attack [52].

In an effort towards understanding the advantage an intruder has by controlling these parameters, and to provide a robust evaluation framework, we present a formal framework to reason about the robustness of an IDS evaluation method. For modeling purposes we decompose the $\mathcal{IDS}$ algorithm into two parts: a *detector* $\mathcal{D}$ and a *decision maker* $\mathcal{DM}$.

For the case of an anomaly detection scheme, $\mathcal{D}(\mathbf{x}[j])$ outputs the anomaly score $y[j]$ on input $\mathbf{x}[j]$ and $\mathcal{DM}$ represents the threshold that determines whether to consider the anomaly score as an intrusion or not. For a misuse detection scheme, $\mathcal{DM}$ has to decide to use the signature to report alarms or decide that the performance of the signature is not good enough to justify its use and therefore will ignore all alarms. An $\mathcal{IDS}$ algorithm is the composition of algorithms $\mathcal{D}$ (an algorithm from where we can obtain an ROC curve) and $\mathcal{DM}$ (an algorithm responsible for selecting an operating point). During operation, an $\mathcal{IDS}$ receives a continuous data stream of event features $\mathbf{x}[1], \mathbf{x}[2], \ldots$ and classifies each input $\mathbf{x}[j]$ by raising an alarm or not.

We next study the performance of an IDS under an adversarial setting. A basic assumption to make is to consider that the intruder knows everything that we know about the environment and can make inferences about the situation the same way as we can. Under this assumption we assume that the base-rate $\hat{p}$ estimated by the IDS, its estimated operating condition $(\hat{P}_{FA}, \hat{P}_D)$ selected during the evaluation, the original *ROC* curve (obtained from $\mathcal{D}$) and the cost function $C(I, A)$ are *public values* (i.e. known to the intruder).

We model the capability of an adaptive intruder by defining some confidence bounds. We assume an intruder can deviate $\hat{p} - \delta_l, \hat{p} + \delta_u$ from the expected $\hat{p}$ value. Based on our confidence in the detector algorithm and how hard we expect it to be for an intruder to evade the detector, we define α and β as bounds to the amount of variation we can expect during the IDS operation from the false alarms and the detection rate (respectively) we expected, i.e. variation from $(\hat{P}_{FA}, \hat{P}_D)$ [87].

The intruder also has access to an oracle *Feature*$(\cdot,\cdot)$ that simulates an event to input into the IDS. *Feature*$(0,\zeta)$ outputs a feature vector modeling the normal behavior of the system that will raise an alarm with probability ζ (or a crafted malicious feature to only raise alarms in the case *Feature*$(0,1)$). And *Feature*$(1,\zeta)$ outputs the feature vector of an intrusion that will raise an alarm with probability ζ.

Definition 1. A $(\delta, \alpha, \beta)-$*intruder* is an algorithm $\mathfrak{I}$ that can select its frequency of intrusions p_1 from the interval $\delta = [\hat{p} - \delta_l, \hat{p} + \delta_u]$. If it decides to attempt an intrusion, then with probability $p_2 \in [0, \beta]$, it creates an attack feature $\mathbf{x}$ that will go undetected by the IDS (otherwise this intrusion is detected with probability $\hat{P}_D$). If it does not attempt an intrusion, with probability $p_3 \in [0, \alpha]$ it creates a feature $\mathbf{x}$ that will raise a false alarm in the IDS. If $\delta_l = p$ and $\delta_u = 1 - p$ we say that $\mathfrak{I}$ has the ability to make a *chosen-intrusion rate attack*.

We now formalize what it means for an evaluation scheme to be robust; i.e. how confident we are that the

IDS will behave during operation similarly to what we assumed in the evaluation.

4.2.1. Robust Expected Cost Evaluation

We start with the general decision theoretic framework of evaluating the expected cost (per input) $\mathbf{E}[C(I,A)]$ for an IDS.

Definition 2. An evaluation method that claims the expected cost of an $\mathcal{IDS}$ is at most r is *robust* against a (δ, α, β)*–intruder* if the expected cost of $\mathcal{IDS}$ during the attack ($\mathbf{E}^{\delta,\alpha,\beta}[C[I,A)]$) is no larger than r, i.e.

$$\mathbf{E}^{\delta,\alpha,\beta}[C[I,A)] = \sum_{i,a} C(i,a)$$
$$\times \Pr[(I,\mathbf{x}) \leftarrow \mathfrak{J}(\delta,\alpha,\beta); A \leftarrow \mathcal{IDS}(\mathbf{x}) : I = i, A = a] \leq r$$

An IDS is better than others if its expected value under the worst performance is smaller than the expected value under the worst performance of other IDSs. Several important questions can be raised by the above framework. In particular we are interested in finding the least upper bound r such that we can claim the evaluation of $\mathcal{IDS}$ to be *robust*. Another important question is how can we design an evaluation of $\mathcal{IDS}$ satisfying this least upper bound? Solutions to these questions are partially based on game theory. We have [22]

Theorem 2. Given an initial estimate of the base-rate $\hat{p}$, an initial *ROC* curve obtained from $\mathcal{D}$, and constant costs $C(I,A)$, the least upper bound r such that the expected cost evaluation of $\mathcal{IDS}$ is robust is given by

$$r = R(0, \hat{P}^{\alpha}_{FA})(1 - \hat{p}^{\delta}) + R(1, \hat{P}^{\beta}_{D})\hat{p}^{\delta} \tag{19}$$

where

$$R(0, \hat{P}^{\alpha}_{FA}) \equiv [C(0,0)(1 - \hat{P}^{\alpha}_{FA}) + C(0,1)\hat{P}^{\alpha}_{FA}] \tag{20}$$

is the expected cost of the $\mathcal{IDS}$ under no intrusion and

$$R(1, \hat{P}^{\beta}_{D}) \equiv [C(1,0)(1 - \hat{P}^{\beta}_{D}) + C(1,1)\hat{P}^{\beta}_{D}] \tag{21}$$

is the expected cost of the $\mathcal{IDS}$ under an intrusion, and $\hat{p}^{\delta}$, $\hat{P}^{\alpha}_{FA}$ and $\hat{P}^{\beta}_{D}$ are the solution to a zero-sum game between the intruder (the maximizer) and the IDS (the minimizer), with a simple and constructive solution.

The proof of this Theorem is very straightforward [22].

We have analyzed a practical example in [22] for minimizing the cost of a chosen intrusion rate attack, that shows the generality of Theorem 2 and also presents a compelling scenario type where a probabilistic IDS makes sense. The example considers an ad hoc network scenario similar to [16,58,94] where nodes monitor and distribute reputation values of other nodes' behavior at the routing layer. The monitoring nodes report selfish actions (e.g. nodes that agree to forward packets in order to be accepted in the network, but then fail to do so) or attacks (e.g. nodes that modify routing information before forwarding it). There is a network operator considering implementing a *watchdog* monitoring scheme to check the compliance of nodes forwarding packets as in [58]. The operator then plans an evaluation period of the method where trusted nodes are the watchdogs reporting the misbehavior of other nodes. Since the detection of misbehaving nodes is not perfect, during the evaluation period the network operator is going to measure the consistency of reports given by several watchdogs and decide if the watchdog system is worth keeping or not. See [22] for details.

4.2.2. Robust IDOC Evaluation

We have also analyzed the robustness of the evaluation with IDOC curves. It is easy to see that the worst attacker for the evaluation is an intruder $\mathfrak{J}$ that selects $p_1 = \hat{p} - \delta_l$, $p_2 = \alpha$ and $p_3 = \beta$.

Corollary 3. For any point $(P\hat{P}V, \hat{P}_D)$ corresponding to $\hat{p}$ in the IDOC curve, a (δ, α, β) – *intruder* can decrease the detection rate and the positive predictive value to the pair $(P\hat{P}V^{\delta,\alpha,\beta}, \hat{P}^{\beta}_{D})$, where $\hat{P}^{\beta} = \hat{P}_D(1 - \beta)$ and

$$P\hat{P}V^{\delta,\alpha,\beta} = \frac{P^{\beta}_{D}p - P^{\beta}\delta}{P^{\beta}_{D}p + P^{\alpha}_{FA}(1-p) + \delta P^{\alpha}_{FA} - \delta P^{\beta}_{D}} \tag{22}$$

5. Distributed Trust Dynamics

In this section we investigate the dynamics of trust 'spreading' in autonomic networks. We find, surprisingly, that several of the analytical formulations appear similar to recent results and formulations for collaborative control and control of formations [2,45,64]. On the other hand, our results for these dynamic trust problems have analogs for collaborative control problems that are not found in the control literature; pointing promising research directions.

We model an autonomous network as a directed graph $G(V, E)$, in which nodes are the entities/peers in the network and links represent trust relations. We call the graph G the *trust graph*, in order to distinguish it from the *physical graph*, in which nodes are connected if they are one hop away in terms of physical transmissions. Suppose that $|V| = N$ and nodes are labeled with indices $V = \{1, \ldots, N\}$.

In a distributed environment, there is no centralized system to manage trustworthiness of entities. However, entities may still rate each other based on their previous interactions. For example, when node i requests files from node j, i may rate j based on whether j replies to his requests and the quality of these files. A directed link from node i to node j in G, denoted as (i,j), corresponds to the *direct* trust relation that entity i has on entity j and the weight on the link represents the degree of confidence i has on j, denoted as $c_{ij} : V \times V \rightarrow [-1, 1]$. $c_{ij} = 1$ represents completely positive confidence i has on j, and $c_{ij} = -1$ represents completely negative confidence. $c_{ij} = 0$ means totally uncertain. Trust relations are asymmetric, so generally $c_{ij} \neq c_{ji}$. Assume nodes' opinions are fixed for the sake of simplicity. We define the neighbor set of node i as

$$\mathcal{N}_i = \{j|(i,j) \text{ or } (j,i) \in E\} \subseteq V \setminus \{i\},$$

which is the set of nodes that are directly connected to i.

Nodes in the network are assumed to be either GOOD or BAD, denoted by $t_i = 1$ or -1 for node i. The vector $T = [t_1, \ldots, t_N]$ is called the *real* trust vector in order to distinguish it from the *estimated* trust vector below. Mathematically speaking, trust evaluation is to estimate the trustworthiness of nodes. Let s_i be the estimated trust value of node i and vector $S = [s_1, \ldots, s_N]$ be the estimated trust vector. If $s_i = 1$, we call node i trusted, which is a subjective concept, while $t_i = 1$ means node i is a good node, which is an existing but unknown fact. The evaluation result is the estimate s_i rather than the real trust value t_i.

Without centralized trusted authority, confidence values may not be able to represent true states of the target even from good nodes. For example, in a network with active attackers, the target node that used to be good may be compromised by bad nodes, or because of communication constraints – the past experience may not completely represent the current behavior of the target. So c_{ij} is modeled as a random variable depending on the real trust values of i and j. The conditional probability $\Pr[c_{ij}|t_i, t_j]$ represents the probability of the c-value being equal to c_{ij} given t_i and t_j. Trivially, if $(i,j) \notin E$, $\Pr[c_{ij} = 0|t_i, t_j] = 1$. We assume that all c-values are independent with each other, i.e.,

$$\Pr[c_{ij}, c_{kl}|t_i, t_j, t_k, t_l] = \Pr[c_{ij}|t_i, t_j] \cdot \Pr[c_{kl}|t_k, t_l],$$

where i, j, k, l may not be distinct.

For a homogeneous distributed network, all nodes are equal. There is no reason to specialize any particular node. Therefore, trust evaluation should take all available trust information into account. Suppose node i is the target of trust evaluation. A natural approach is to aggregate all its neighbors' opinions. This is called a *local voting rule (local policy)*, in which votes are neighbors' c-values on the target. However, a rule using naive summation is not a good estimate, because of the following reasons:

1. Trustworthiness of voters: Opinions from nodes with high (estimated) trust values are more credible, so they should carry larger weights. So the voting rule should be a weighted sum.
2. Conflicting opinions between the target and voters: Suppose j is one of the voters of target i and their opinions on each other are conflicting, say $c_{ij} = 1$, while $c_{ji} = -1$. In order to mitigate the effect of such conflicting votes, we use *effective* votes, with

$$\hat{c}_{ji} = c_{ji} + \alpha c_{ij} \tag{23}$$

where α is a constant. For simplicity we set $\alpha = 1$.

Based on the above arguments, the local voting rule is

$$s_i = f(\hat{c}_{ji} s_j | j \in \mathcal{N}_i) \tag{24}$$

where $f : R \rightarrow [-1, 1]$. The trust value s_j depends on trust values of j's neighbors and their votes on j. Notice that s_j is also evaluated at the same time, and so are j's neighbors. The whole evaluation therefore evolves as local interactions iterate throughout the network and Eq. (24) becomes

$$s_i(k+1) = f\left(\hat{c}_{ji} s_j(k) | j \in N_i\right). \tag{25}$$

Thus trust evaluation is a dynamic process which evolves with time.

Our interest is to study the evolution of the estimated trust vector S and its values at equilibrium. The motivation for trust management is to be able to detect bad nodes and trust good nodes. It is important to investigate whether S can correctly estimate the trust vector T at steady state.

5.1. A Stochastic Threshold Rule

Guided by the voting rule in Eq. (25), we have designed [48] a specific evaluation rule for analysis. The target node is either trusted or distrusted as decided by the threshold rule

$$s_i(k+1) = \begin{cases} 1, & \text{if } m_i(k) \geq \eta \\ -1, & \text{if } m_i(k) < \eta \end{cases}, \tag{26}$$

where

$$m_i(k) = \sum_{j \in \mathcal{N}_i} \hat{c}_{ji} s_j(k) \tag{27}$$

is the weighted sum of the votes from i's neighbors.

However, as we have discussed, uncertainty of opinions by peers is inevitable for autonomous networks. Thus we introduce randomness into our rule. Obviously, if the weighted sum m_i is large, s_i will take value 1 with high probability and vice versa. If m_i is right on the threshold η, it should choose 1 or -1 with equal probability. So our stochastic threshold rule is defined as:

$$\Pr[s_i(k+1) = 1 | m_i(k)] = \frac{e^{b(m_i(k)-\eta)}}{Z_i(k)} \tag{28}$$

$$\Pr[s_i(k+1) = -1 | m_i(k)] = \frac{e^{-b(m_i(k)-\eta)}}{Z_i(k)} \tag{29}$$

where $Z_i(k)$ is the normalization factor

$$Z_i(k) = e^{b(m_i(k)-\eta)} + e^{-b(m_i(k)-\eta)}, \tag{30}$$

and $b > 0$ is a constant representing the degree of certainty. A small b represents a highly uncertain scenario. By placing the value of $s_i(k+1)$ into the right hand sides of both Eqs (28) and (29), the stochastic voting rule can be combined into one formula

$$\Pr[s_i(k+1) | m_i(k)] = \frac{e^{bs_i(k+1)(m_i(k)-\eta)}}{Z_i(k)}. \tag{31}$$

Our evaluation rule is essentially an updating rule. In the autonomous environment, it is very difficult to achieve synchrony. Thus the system should only use asynchronous updates. The probability that node i is chosen as the target is q_i, and $\sum_{i \in V} q_i = 1$.

5.1.1. Convergence

We have from [48]

Theorem 4.

For the stochastic voting rule defined by Eq. (31) and using random asynchronous updates, if $b \in (0, \infty)$ and $q_i > 0, \forall i \in V$, we have that

(a) the voting rule converges to the steady state with a unique stationary distribution;
(b) the distribution $\pi_S = \frac{e^{bU(S)}}{Z}$ is the unique stationary distribution.

Having derived the stationary distribution, we are able to compute the probability of correct estimation. Let vector SS be equal to the vector S at steady state. Then the probability of correct estimation, including trusting good nodes and detecting bad nodes, is [48]

$$P_{\text{correct}} = \{\text{Expected } \# \text{ of } SS_i = T_i\}$$
$$= \mathbf{E}\left[1 - \frac{\|SS - T\|_1}{2N}\right].$$

where $\|SS - T\|_1 = \sum_{i \in V} |SS_i - T_i|$.

The stationary distribution

$$\pi_S = \frac{e^{bU(S)}}{Z}$$

is called *Gibbs distribution*. The Gibbs distribution is closely related to the local interactions of our voting rule.

5.1.2. Markov Random Field

Re-write Eq. (31) with a small modification: replace m_i with $s_j(k), j \in \mathcal{N}_i$.

$$\Pr[s_i(k+1) | S(k)] = \Pr[s_i(k+1) | s_j(k), j \in \mathcal{N}_i]. \tag{32}$$

Equation (32) in fact presents a Markov type property, i.e., the probability of the estimated trust value for a certain node i, s_i, given the estimated trust values of all the other nodes in the network, is the same as the probability of s_i, given only the estimated trust values of the neighbors of i. As opposed to a Markov chain, which has the Markov property with respect to time, Eq. (32) displays the Markov property in space. A distribution with such a property is called a *Markov random field* (MRF) [51]. The well-known Hammersley-Clifford theorem [51] proves the equivalence between a MRF on a graph and the Gibbs distribution.

5.2. Trust at Steady State

We have investigated properties of the estimated trust values when the voting rule reaches the steady state. At first, we introduce an important model that models local interactions of magnets in physics – the Ising model.

5.2.1. Ising Model and Spin Glasses

The Ising model [92] describes the interaction of magnetic moments or 'spins' of particles, where some particles seek to align with one another (ferromagnetism), while others try to anti-align (antiferromagnetism). In the Ising model, s_i is the orientation of the spin at particle i. $s_i = 1$ or -1 indicates the spin at i is 'up' or 'down' respectively. A Hamiltonian, or energy, for a configuration S is given by

$$H(S) = -\sum_{(i,j)} J_{ij} s_i s_j - mH \sum_{i} s_i. \tag{33}$$

The first term represents the interaction between spins. The second term represents the effect of the external (applied) magnetic field. Then the probability of configuration S is

$$\Pr[S] = \frac{e^{-\frac{1}{kT}H(S)}}{Z}, \tag{34}$$

where T is the temperature and k is the Boltzmann constant. In the Ising model, the local interaction 'strengths' J_{ij}'s are all equal to a constant J, which is either 1 or -1. In recent years, an extension of the Ising model called the Edwards-Anderson model of spin glasses is used to study local interactions with independently random J_{ij} [63], which correspond to the c_{ij} in our voting rule.

5.2.2. *Virtuous Network*

Now let us go back to our discussion of trust at steady state. We start with the simplest case: a virtuous network, where all nodes are good and they always have full confidence on their neighbors, so $t_i = 1$, $\forall i \in V$ and $c_{ij} = 1, \forall (i,j) \in E$. Then the stationary distribution π is exactly the same as the one in the Ising model with

$$b = \frac{1}{2kT} \quad \text{and} \quad \eta = -\frac{mH}{kT}. \tag{35}$$

Since all nodes are good with $t_i = 1$ and SS_i is either 1 or -1, the probability of correct estimation can be written as

$$P_{\text{correct}} = \frac{\mathrm{E}[\langle SS \rangle] + N}{2N},$$

where $\langle SS \rangle = \sum_{i \in V} SS_i$. In the terminology of physics, $\langle SS \rangle$ is called the total magnetization. It is known that when the external field $H > 0$, $\mathrm{E}[\langle SS \rangle]$ is positive and when $H < 0$, it is negative. According to (35), the threshold $\eta > 0$ corresponds to $H < 0$, thus $\mathrm{E}[\langle SS \rangle] < 0$ and $P_{\text{correct}} < 0.5$. Similarly when η is negative, $P_{\text{correct}} > 0.5$.

We used simulations to study the value of P_{correct} with respect to parameters η and b. The network topology for all the simulations is a two-dimensional lattice with periodic boundary. The number of nodes is 100 and each takes four nearest nodes as neighbors. We chose the lattice because most theoretical results for the Ising model are for the 2-D lattice. In Sec. V-B.3, we discuss the effect of network topology.

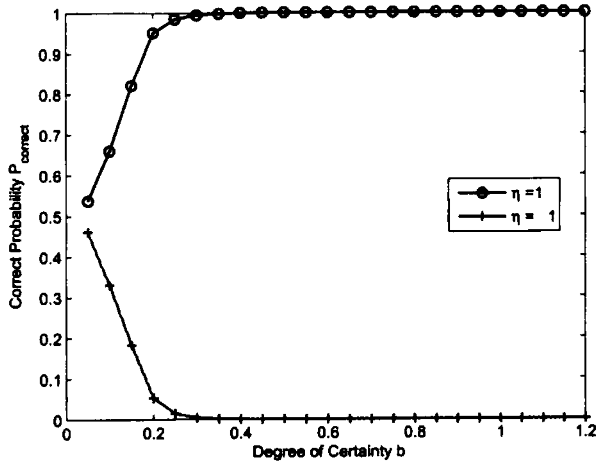

Fig. 8. P_c vs. b with $\eta < 0$ and $\eta > 0$.

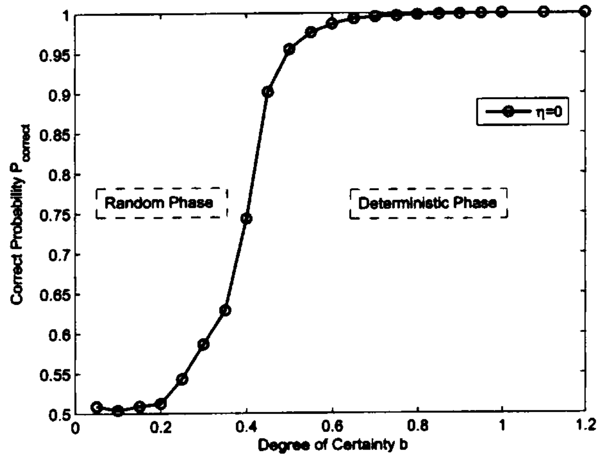

Fig. 9. P_{correct} vs. b with $\eta = 0$.

Figures 8 and 9 represent the probability of correct estimation as a function of b for η being negative, positive or zero. For $\eta > 0$ (the rule is chosen to be conservative) the probability of correct estimation is less than half. On the other hand, for $\eta = 0$ or $\eta < 0$, if the value b is properly chosen ($b > 0.6$), P_{correct} is close to 1. Therefore, the threshold η must be non-positive.

The other interesting property is the phase transition phenomenon observed in Fig. 9 when b is in [0.4,0.5]. Phase transitions have been extensively studied by physicists in the Ising model. If we look closer into the interval when $b < 0.4$, the estimated trust value of each node is changing all the time and looks like random jumping. When b is above the critical

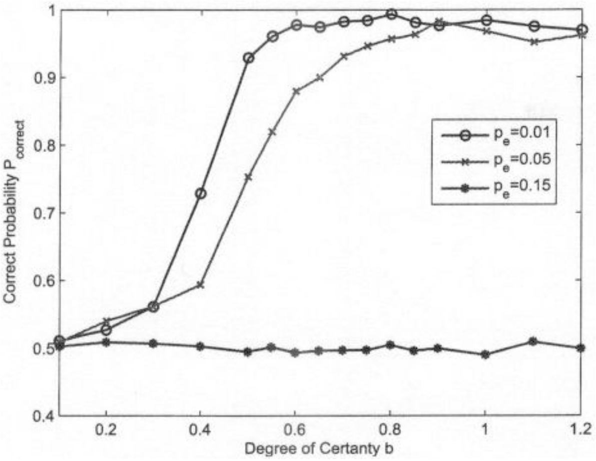

Fig. 10. P_c vs. b with link errors p_e. $\eta = 0$.

value, all values converge steadily to 1. We call the first interval the *random phase*, the second the *deterministic phase*.

The discovery of phase transitions in our voting rule is quite surprising given that the rule itself is very simple. More importantly, the fact that a small change in the parameter might result in a totally opposite performance of our voting rule proves the necessity of doing careful analysis before applying any distributed algorithms.

As discussed, due to uncertainty and incompleteness of trust evidence, c_{ij} should be modeled as a random variable rather than being always 1. Let's assume $c_{ij} \in \{-1, 0, 1\}$ and define the probability that a good node has an incorrect opinion on its neighbors as p_e, then we have

$$p_e = \Pr\left[c_{ij} \neq t_j | t_i = 1\right] \text{for all } i \text{ good.}$$

Thus in a virtuous network, the distribution of c_{ij} is

$$\Pr(c_{ij} = 1) = 1 - p_e; \quad \Pr(c_{ij} = -1) = p_e. \quad (36)$$

We again investigate the phase transition. As shown in Fig. 10, the phase transition still happens when $\eta = 0$. However, as p_e increases, the wrong votes with value -1 gradually destabilize the votes of value 1. Thus it is harder to keep s_i's equal to 1, which means that b_c becomes larger and the system more probably stays in the random phase given a high link error p_e. When p_e is large enough ($p_e = 0.15$), as shown, the system always stays in random phase.

In [63], the authors theoretically studied phase transitions between random and deterministic phases, and introduced the replica symmetry method to solve them analytically. Based on this method, very good approximations of values, such as $\mathbf{E}[\langle SS \rangle]$ and $\mathbf{E}[SS_i^2]$, can be derived. The mathematical manipulation of the replica symmetry method is beyond the scope of this paper, but it is definitely a very good direction for future research. Explicit expressions for these values will provide even better guide for network management.

5.2.3. Network Topology

We have shown [9–11,46] that network topology has significant influence on trust evaluation. The network model we used is the small-world model, which has its roots in social systems [59]. In the past five years, there has been substantial research on the small-world model in various complex networks, such as Internet and biological systems [91]. In [17], it was shown that the PGP certificate graph has the small-world property.

Several small-world models have been proposed. In our work we used the small-world model proposed by Watts and Strogatz in [91] (WS model), because it is relatively simple but retains the fundamental properties of practical networks. In the WS model, we start from a ring lattice with $N = 100$ nodes and degree of each node $k = 4$. Each edge is rewired at random so as to create shortcuts with the percentage of shortcuts being the parameter P_{rw}. This construction models the graph transition between regular lattices ($P_{rw} = 0$) and chaotic random graphs ($P_{rw} = 1$).

Our simulation results are shown in Fig. 11, with different shortcut percentages P_{rw}. We observe that the performance improves as the model changes from regular lattices to random graphs. For instance, as $b = 0.4$, $P_{\text{correct}} = 0.55$ for regular lattices, while $P_{\text{correct}} = 0.85$ for random graphs. In particular, the most obvious improvement happens when P_{rw} increase from 0.01 to 0.1, which corresponds to the small-world topology. Therefore, a few shortcuts in the network greatly improve the performance of the trust evaluation rule.

Clearly the network topology has great influence on the system performance. As future research, it is interesting to study our trust evaluation rule under real trust network topologies, and to investigate what kind of network topology has the best performance in terms of trust evaluation.

Although from a certain perspective stochastic policies (or voting rules) may be the most appropriate for the uncertain environment of wireless autonomous networks, deterministic policies are also employed. They lead to an even more transparent similarity with problems recently considered in formation control [45]. There are several choices for such deterministic policies or voting rules. For instance, it can be the

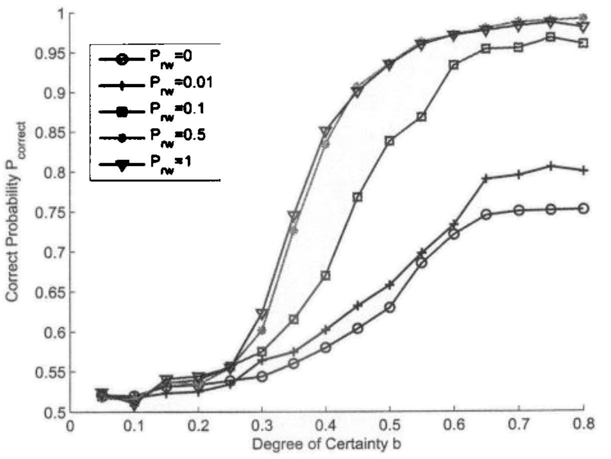

Fig. 11. The effect of network topology. P_{rw} is the percentage of shortcuts in WS small world model. The network with $P_{rw} \in [0.01, 0.1]$ has the small world property. $p_e = 0.1$.

average, maximum or minimum of all votes. A rule we have used [9–11,46], updates the weighted average of all neighbor votes followed by a threshold rule on the steady state of trust values (defined as $s_i = \lim_{n\to\infty} s_i(n)$). This rule dependents on a system parameter η as follows:

$$\text{Node } i \text{ is} \begin{cases} \text{trusted,} & \text{if } s_i \geq \eta \\ \text{neutral,} & \text{if } s_i < \eta \end{cases}.$$

It can be interpreted as 'local majority vote'.

In [9,11,46] we investigated the dynamics of 'trust spreading' and in particular the time it takes to reach the steady state, in other words how fast the trust values converge. Using Perron-Frobenius theory [15], we investigated and answered the question: what kind of networks or which network topology induces faster convergence? Small world models have two prominent properties: high clustering coefficient and small average graph distance between node pairs. The latter is the reason for faster convergence [9,11,46]. Our examples demonstrate [9,11,46], that even the addition of just 1% of the total edges, as shortcuts, induces the convergence time to drop from 5000 rounds to 500 rounds – a ten-fold decrease! Thus trust is established much faster in a small-world network than in a regular lattice.

6. Cooperative Games in Networks

We have investigated cooperative games [32,35,75] in networks in order to establish fundamental principles for node collaboration in autonomous networks [9,11,47,49]. Nodes play cooperative games with their neighbors iteratively. At each time step, two neighboring nodes only play the game once. Cooperative games [35] are normally represented in *characteristic function form* which is a finite set $N = \{1, \ldots, N\}$, the set of players, and a function (characteristic function) $v : 2^N \to \mathbf{R}$ defined on all subsets (*coalitions*) of N with $v(\emptyset) = 0$. We denote such a game as $\Gamma = (N, v)$. Define S, a subset of N, as a coalition if all nodes in S cooperate. Then $v(S)$ is interpreted as the maximum utility (payoff) S can get without the cooperation of the rest of the players $N \setminus S$. In order to simplify our analysis, we assumed the payoff only depends on the interacting two parties. Suppose x_{ij} is the payoff of i from the game between i and j. Since games are played on networks, $x_{ij} \neq 0$ only if i and j are neighbors.

The characteristic function of our cooperative game is the summation of the payoffs from all cooperative pairs

$$v(S) = \sum_{i,j \in S} x_{ij} \tag{37}$$

Based on this model, we have investigated stable solutions for enforcing cooperation among nodes, and proved two efficient methods: negotiation and trust [9,11,47,48].

The main concern in cooperative games [35] is how the total payoff from a partial or complete cooperation of the players is divided among the players. A *payoff allocation* is a vector $x = (x_i)_{i \in N}$ in $\mathbf{R}^N$, where each component x_i is interpreted as the payoff allocated to player i. We say that an allocation x is *feasible for a coalition* S iff $\sum_{i \in S} x_i \leq v(S)$. When we think of a reasonable and stable payoff, the first thing that comes to mind is a payoff that would give each coalition at least as much as the coalition could enforce itself without the support of the rest of the players. In this case, players couldn't get better payoffs if they form separated coalitions than from the grand coalition N. The set of all these payoff allocations of the game $\Gamma = (N, v)$ is called its *core*, denoted $C(\Gamma)$, and is the set of all n-vectors x satisfying:

$$x(S) \geq v(S) \quad \forall S \subset N, \tag{38}$$

$$x(N) = v(N), \tag{39}$$

where $x(S) = \sum_{i \in S} x_i$ for all $S \subset N$.

Trust is a useful incentive for encouraging nodes to collaborate. Nodes who refrain from cooperation get lower trust value and will be eventually penalized because other nodes tend to only cooperate with highly trusted ones. The trust values of each node will eventually influence its payoff. Let's assume, for node i, the loss of not cooperating with node j is a nondecreasing function of x_{ji}, because the more j loses, the more effort j takes to reduce the trust value of i. Denote the loss for i being noncooperative with j as $l_{ij} = f(x_{ji})$ and $f(0) = 0$. For simplicity, assume the characteristic function is a linear combination of the original payoff and the loss, which is shown as

$$v'(S) = \sum_{i,j \in S} x_{ij} - \sum_{i \in S, j \notin S} f(x_{ji}) \tag{40}$$

The new game v' is denoted as $\Gamma'(N, v')$.

Theorem 5. If $\forall i,j,\ x_{ij} + f(x_{ji}) \geq 0,\ C(\Gamma') \neq \emptyset$ and $x_i = \sum_{j \in N} x_{ij}$ is a point in $C(\Gamma')$.

For the proof we refer to [11]. Thus we have showed that by introducing a trust mechanism, all nodes collaborate with their neighbors without any negotiation. We further investigated the evolution of this iterated cooperative game in [11]. We designed a game evolution algorithm, where nodes decide to cooperate with their neighbors based on their payoffs and the trustworthiness of their neighbors at the previous time instant. We showed that under certain simple conditions on the forgiveness of nodes, the iterated game converges to Nash equilibrium [11]. These results are examples of constrained cooperative games, or *constrained coalitional games*, [32,75], which provide an excellent framework for investigating the fundamental tradeoffs between the benefits of cooperation vs. the cost of cooperation in autonomic networks; see our work in [47,49].

We have also performed simulation experiments with our evolution algorithm. In the simulations, we didn't assume the condition that $\forall i, x_i > 0$, instead the percentage of negative links is the simulation parameter. We can report that without this condition, our iterated game with the trust scheme can still achieve very good performance. Figure 12 shows that cooperation is highly promoted under the trust mechanism.

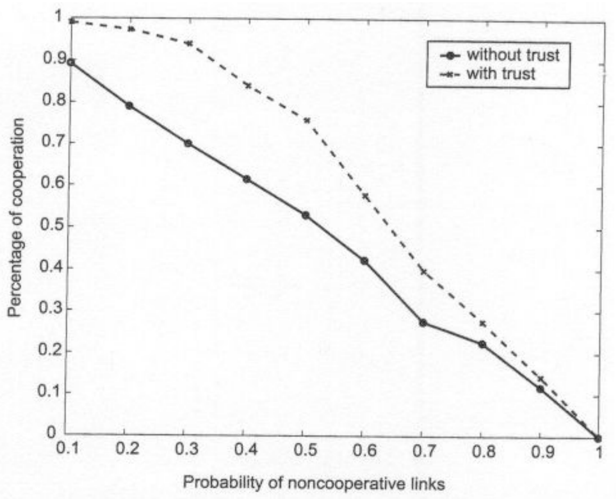

Fig. 12. Percentage of cooperation among nodes in steady states vs. initial percentage of non-cooperative links.

7. Semiring-Based Trust Evaluation Metrics

In this section we view the trust inference problem as a generalized shortest path problem on a weighted directed graph $G(V, E)$ (*trust graph*) [85]. The vertices of the graph are the users/entities in the network. A weighted edge from vertex i to vertex j corresponds to the *opinion* that entity i, also referred to as the *issuer*, has about entity j, also referred to as the *target*. The weight function is $l(i,j) : V \times V \longrightarrow S$, where S is the opinion space.

Each opinion consists of two numbers: the *trust* value, and the *confidence* value. The former corresponds to the issuer's estimate of the target's trustworthiness. For example, a high trust value may mean that the target is one of the good users, or that the target is able to give high quality location information, or that a digital certificate issued for the target's public key is believed to be correct. On the other hand, the confidence value corresponds to the accuracy of the trust value assignment. A high confidence value means that the target has passed a large number of tests that the issuer has set, or that the issuer has interacted with the target for a long time, and no evidence of malicious behavior has appeared.

Nodes assign their opinions based on local observations. For example, each node may be equipped with a mechanism that monitors neighbors for evidence of malicious behavior, as in [58]. Alternatively, two users may come in close contact and visually identify each other, or exchange public keys, as suggested in [18]. In any case, the input to the system is local: however, extant pieces of evidence based on, e.g., previous interactions with no longer neighboring nodes can also be taken into account for the final decision. This would come into play when two nodes that have met in the past need now to make a trust decision for each other. Of course, the confidence value for such evidence would diminish over time. One consequence of the locality of evidence gathering is that the trust graph initially overlaps with the physical topology graph: The nodes are obviously the same, and the edges are also the same if the trust weights are not taken into account. As nodes move, opinions for old neighbors are preserved, so the trust graph will have more edges than the topology graph. As time goes by, these old opinions fade away, and so do the corresponding edges.

In the framework described [81,84,85], two versions of the trust inference problem can be formalized. The first is finding the trust-confidence value that a source node A should assign to a destination node B, based on the intermediate nodes' trust-confidence values. Viewed as a generalized shortest path problem, it amounts to finding the generalized distance between nodes A and B. The second is finding the most trusted path between nodes A and B. That is, find a sequence of nodes $\langle v_0 = A, v_1, \ldots, v_k = B\rangle : (v_i, v_{i+1}) \in E$, $0 \leq i \leq k-1$ that has the highest aggregate trust value among all trust paths starting at A and ending at B.

The core of our approach is the two operators used to combine opinions: One operator (denoted $\otimes$) combines opinions along a path, i.e. A's opinion for B is combined with B's opinion for C into one indirect opinion that A should have for C, based on B's recommendation. The other operator (denoted $\oplus$) combines opinions across paths, i.e. A's indirect opinion for X through path p_1 is combined with A's indirect opinion for X through path p_2 into one aggregate opinion. Then, these operators can be used in a general framework for solving path problems in graphs, provided they satisfy certain mathematical properties, i.e. form an algebraic structure called a semiring.

7.1. Semirings for Trust

For a more complete survey of the issues briefly exposed here, see Rote [72].

A *semiring* is an algebraic structure $(S, \oplus, \otimes)$, where S is a set, and $\oplus, \otimes$ are binary operators with the following properties ($a, b, c \in S$):

- $\oplus$ is commutative, associative, with a neutral element $\mathbb{0} \in S$.
- $\otimes$ is associative, with a neutral element $\mathbb{1} \in S$, and $\mathbb{0}$ as an absorbing element.
- $\otimes$ distributes over $\oplus$.

A semiring $(S, \oplus, \otimes)$ with a partial order relation $\preceq$ that is monotone with respect to both operators is called an *ordered semiring* $(S, \oplus, \otimes, \preceq)$:

$$a \preceq b \text{ and } a' \preceq b' \Longrightarrow a \oplus a' \preceq b \oplus b'$$
$$\text{and } a \otimes a' \preceq b \otimes b'$$

Based on intuitive concepts about trust establishment, we can expect the binary operators to have certain properties in addition to those required by the semiring structure.

Since an opinion should deteriorate along a path, we require the following for the $\otimes$ operator ($a, b \in S$):

$$a \otimes b \preceq a, b$$

where $\preceq$ is the partial order relation.

Regarding aggregation across paths with the $\oplus$ operator, we generally expect that opinion quality will improve, since we have multiple opinions. If the opinions disagree, the more confident one will weigh heavier. Similar to the $\otimes$ operator, we require that the $\oplus$ operator satisfies ($a, b \in S$):

$$a \oplus b \succeq a, b$$

7.1.1. Path Semiring

In our first semiring [81,85], the opinion space is $S = [0, 1] \times [0, 1]$. Our choices for the $\otimes$ and $\oplus$ operators are as follows:

$$(t_{ik}, c_{ik}) \otimes (t_{kj}, c_{kj}) = (t_{ik}t_{kj}, c_{ik}c_{kj}) \tag{41}$$

$$(t_{ij}^{p_1}, c_{ij}^{p_1}) \oplus (t_{ij}^{p_2}, c_{ij}^{p_2}) = \begin{cases} (t_{ij}^{p_1}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} > c_{ij}^{p_2} \\ (t_{ij}^{p_2}, c_{ij}^{p_2}) & \text{if } c_{ij}^{p_1} < c_{ij}^{p_2} \\ (t_{ij}^{*}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} = c_{ij}^{p_2} \end{cases} \tag{42}$$

where $(t_{ij}^{p_1}, c_{ij}^{p_1})$ is the opinion that i has formed about j along the path p_1, and $t_{ij}^{*} = \max(t_{ij}^{p_1}, t_{ij}^{p_2})$.

Since both the trust and the confidence values are in the $[0, 1]$ interval, they both decrease when aggregated along a path. When opinions are aggregated across paths, the one with the highest confidence prevails. If the two opinions have equal confidences but different trust values, we pick the one with the highest trust value. We could have also picked the lowest trust value; the choice depends on the desired semantics of the application.

This semiring essentially computes the trust distance along the most confident trust path to the destination.

7.1.2. Distance Semiring

Our second proposal [81,85], the distance semiring, is based on the *Expectation semiring* of Eisner [33], used for speech/language processing:

$$(a_1, b_1) \otimes (a_2, b_2) = (a_1b_2 + a_2b_1, b_1b_2)$$
$$(a_1, b_1) \oplus (a_2, b_2) = (a_1 + a_2, b_1 + b_2)$$

The opinion space is $S = [0, \infty] \times [0, 1]$. Before using this semiring, the pair (trust, confidence)= (t, c) is mapped to the weight $(c/t, c)$. The motivation for this mapping becomes clear when we describe its effect on the results of the operators. The binary operators are then applied to this weight, and the result is mapped back to a (trust, confidence) pair. For simplicity, we only show the final result without the intermediate mappings.

$$(t_{ik}, c_{ik}) \otimes (t_{kj}, c_{kj}) \rightarrow \left(\frac{1}{\frac{1}{t_{ik}} + \frac{1}{t_{kj}},} c_{ik} c_{kj} \right)$$

$$\left(t_{ij}^{p_1}, c_{ij}^{p_1}\right) \oplus \left(t_{ij}^{p_2}, c_{ij}^{p_2}\right) \rightarrow \left(\frac{c_{ij}^{p_1} + c_{ij}^{p_2}}{\frac{c_{ij}^{p_1}}{t_{ij}^{p_1}} + \frac{c_{ij}^{p_2}}{t_{ij}^{p_2}}}, c_{ij}^{p_1} + c_{ij}^{p_2} \right)$$

So, when aggregating along a path, both the trust and the confidence decrease. The component trust values are combined like parallel resistors. When aggregating across paths, the total trust value is the weighted harmonic average of the components, with weights their confidence values.

The algorithm of Mohri [60], computes the ⊕-sum of all path weights from a designated node s to all other nodes in the trust graph $G = (V, E)$. This is an extension to Dijkstra's algorithm [29]. The crucial parameter of the topology is the number of paths from the source to the other nodes [60]. So, the more sparse the network, the more efficient the algorithm. The algorithm can be executed in a distributed fashion with local data exchanges only.

Thus what we want to compute is the following semiring-summation over all the paths p from s to d.

$$t_{sd} = \bigoplus_{p} t_{sd}^{p}$$

We can break up these paths according to their last link:

$$t_{sd} = \bigoplus_{k \in N_d} t_{sk} \otimes w(k, d)$$

where N_d are the in-neighbors of d: the users that have a direct opinion about d. If we now let d vary over the set of all users, t_{sd} becomes a vector, and we can write:

$$\vec{t} = \vec{t}W \tag{43}$$

where W is the matrix of direct opinions. So, the result of the trust computation for User s (s's indirect opinions about everybody else), is the eigenvector of W associated with ① (the neutral element for ⊗, which in our semiring is also the maximum element). So, we have formulated the problem of computing indirect opinions as an eigenvector problem. Perron-Frobenius theory for semirings (see e.g. Baccelli, Cohen, Olsder, and Quadrat [6, Thm 3.23]) tells us that if W is irreducible (i.e. the graph is strongly connected, which we assume here), then there exists exactly one eigenvalue, but possibly many eigenvectors. This eigenvalue λ is equal to the maximum mean circuit weight of the graph.

7.2. Attack Resistance

Suppose now that there exists an Attacker who wants to manipulate the trust computation, i.e. cause User s to compute false opinions about others. The Attacker can change the opinion on a single edge, which would amount to tricking a user into issuing a false opinion, or creating a forged opinion [84]. We want to see what is the maximum damage the Attacker can cause. This is equivalent to asking what single entry change in the matrix W causes the largest change in the eigenvector. The Attacker causes W to become W^*, so t becomes t^*. The damage is equal to $\|t - t^*\|$, where $\|\cdot\|$ is a suitable (e.g. the L_1 or the L_∞) norm.

In what follows [84] we limit our attention to a particular pair s-d. We will examine which edge the Attacker will attack, and characterize the resilience of the s-d trust computation to such single edge attacks.

The problem just described is very similar to computation of tolerances for edges of a network. In short, if p^* is an optimal path from s to d, the upper (lower) tolerance of an edge e with respect to p^* is the largest (smallest) weight of edge e that preserves the optimality of p^*. The most vital edge is defined as the edge that, if deleted, causes the greatest deterioration in the optimal path weight. We have [84]

Theorem 6 (Semiring Tolerances). Let OPT^* be the set of ⊕-optimal paths in $G = (V, E)$. Instead of lower and upper tolerances, α_e and β_e now mean ⊕-minimal and ⊕-maximal values of an edge $e \in E$ that preserve the ⊕-optimality of some path in OPT^*.

1. If $\exists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \left(\bigoplus_{\substack{p : s \rightsquigarrow d \\ w(e) \leftarrow \textcircled{0}}} w(p) \right) \oslash w(p^* \setminus e)$. Moreover, if $\exists p^* \in OPT^* : e \notin p^*$, then $\alpha_e = \textcircled{0}$.
   - $\beta_e = \textcircled{1}$.
2. If $\nexists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \textcircled{0}$. It suffices that $\exists p^* \in OPT^* : e \notin p^*$.
   - $\beta_e = w(p^*) \oslash \left(\bigoplus_{\substack{p : s \rightsquigarrow d \\ w(e) \leftarrow \textcircled{1}}} w(p) \right)$

The operator ⊘ is the inverse of ⊗. Since we are dealing with semirings, ⊘ may not always be defined, as in the case of ⊗ = min. In these cases, $a = b \oslash c$

means that a,b, and c are such that the equality $a \otimes c = b$ holds. Theorem 6 holds for the two specific problems mentioned above.

The benefit of this generalization is that we can directly apply it to semirings where $\oplus$ is max or min, i.e. where there is some optimization involved. Our path semiring is $(\oplus, \otimes, ⓪, ①) = (\max, \cdot, 0, 1)$, so we can directly apply Theorem 6. Lower tolerance is α_e, upper tolerance is β_e.

1. If $\exists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \frac{w(e)}{w(p^*)} \cdot \left(\max_{w(e) \leftarrow 0} w(p)\right)$. Moreover, if $\exists p^* \in OPT^* : e \notin p^*$, then $\alpha_e = 0$.
   - $\beta_e = 1$.

2. If $\nexists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = 0$. It suffices that $\exists p^* \in OPT^* : e \notin p^*$.
   - $\beta_e = \frac{w(p^*)}{\max_{w(e) \leftarrow 1} w(p)}$

If the user d, for which s is computing the indirect opinion, is a good user, then the Attacker will want to reduce the computed opinion. In that case, the link to be attacked is the one with the smallest lower tolerance α_e. The attack will consist of setting the weight of the edge at 0. If, on the other hand, d is a bad user, then the Attacker will try to increase the computed indirect opinion. So, he will attack the edge with the largest upper tolerance, and set its weight to 1. For detailed results see [84].

8. Cooperation and Reputations in Autonomic Networks

With each network, there is an associated protocol which is the way that the network is supposed to operate. The users can choose between participating in the operation of the network or not and, if yes, to what degree (all the time? some of the time?). There are usually pros and cons from the viewpoint of the individual user, so, in general some will decide to participate and some not. For example, in the case of a wireless ad-hoc network, participation means forwarding other users' packets. The incentive is the expectation the user has: Namely, that other users will also forward his packets. The disincentive is that the very action of transmitting data reduces a user's available energy, which is scarce in such networks. Moreover, the user wastes his bandwidth, which he could use to forward his own data.

Note that the benefit of cooperation is somewhat more abstract, global, and indirect than the cost. So, it could be argued that some of the Good users may behave selfishly, and as a result will not take very seriously the incentive that a globally desirable outcome presents. But we consider that users are either Good or Bad, not selfish or unselfish. In particular, all Good nodes behave equally unselfishly in the sense that, in principle, they value the network benefit more than their individual cost. This will be further explained in the discussion on the user model (Section 8.1), but it should not be taken to mean that Good users unconditionally cooperate. If, for instance, none of a Good user's neighbors cooperate (e.g. they do not forward his packets), then the Good user will stop cooperating despite being Good.

8.1. Malicious and Legitimate User Model

The network is modeled as an undirected graph $G = (V, L)$, where each node in V corresponds to one user. An edge $(i,j) \in L$ means that there is a communication link between the users corresponding to nodes i and j. The neighbors of user i, denoted N_i, is the set:

$$N_i = \{j \in V | (i,j) \in L\}. \tag{44}$$

We denote the set of Malicious (Bad) users by V_B, and the set of Benign (Good) users by V_G. It holds that $V_B \cap V_G = \emptyset$ and $V_B \cup V_G = V$. We will be using the term *type* of a player for this property (being Good or Bad).

Users have a choice [82,83,86] between two actions: C (for Cooperate), and D (for Defect). When all users choose their actions, each user receives a payoff that depends on three things: His own action, his neighbors' actions, and his own type (but not his neighbors' types). The payoff is decomposed as a sum of payoffs, one for each link. Each term of the sum depends on the player's own action (which is the same for each link that the player is part of), and the action and type of his neighbor along that link. The payoff of user i is denoted by $R_i(a_i|t_i)$, when i's action is a_i and i's type is t_i. We extend and slightly abuse this notation to denote by $R_i(a_i a_j|t_i)$ the payoff for i when j is a neighbor of i and j's action is a_j. So, the decomposition of i's payoff can be written as:

$$R_i(a_i|t_i) = \sum_{j \in N_i} R_i(a_i a_j|t_i) \tag{45}$$

We assume there are no links between any two Bad users. The Bad users are supposed to be able to communicate and coordinate perfectly; hence, there is no need to restrict their interaction by modeling it in these terms. Moreover, the Bad users know exactly both the topology and the type of each user in the network. Good users only know their local topology,

		Bad	
		C	D
Good	C	$N-E, E-N$	$-E, E$
	D	$0, -E$	$0, 0$

		Good	
		C	D
Good	C	$N-E, N-E$	$-E, 0$
	D	$0, -E$	$0, 0$

Fig. 13. The two games that can take place on a link: Good vs. Bad and Good vs. Good.

e.g., how many neighbors they have and what each one of them plays, but not their types.

The payoffs are shown in table form in Fig. 13 for the two pairs of types that can arise (Good versus Good, and Good versus Bad). We consider [82,83,86] that the game is played repeatedly with an infinite horizon, and time is divided in rounds $t = 1, 2, 3, \ldots$. Actions and payoffs of round t are denoted with a superscript t: a_i^t and R_i^t. The objective of the players in a repeated game is to maximize a function of the sequence of payoffs that they accumulate over the infinite course of the game. We have considered [82,83,86] the average of the payoffs to be the payoff for the whole game:

$$R_i = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} R_i^t.$$

In repeated games, the players are allowed to have full or partial memory of the past actions. Here, we allow the Bad users to have all information about the past (their own moves, as well as everybody else's moves since the first round). On the other hand, the Good users follow a *fictitious play* process, that is, they assume that each of their neighbors chooses his actions according to a fixed probability distribution (Bernoulli in this case, since there are only two actions available: C and D). So, at each round they are choosing the action that maximizes their payoff given the estimates they have for each of their neighbors' strategies. We denote by q_j^t the estimated probability that j will play C in round $t+1$, based on j's actions in rounds $1, \ldots, t$.

Assume that t rounds have been completed, and Good user i is contemplating his move in round $t+1$. His expected payoff for each of the two actions is [82]

$$\begin{aligned} \mathrm{E}R_i(C|G) &= N \cdot \sum_{j\in N_i} q_j^t - |N_i| \cdot E \\ \mathrm{E}R_i(D|G) &= 0. \end{aligned} \qquad (46)$$

So, in order to decide what to play, user i has to compare the expected payoff that each action will bring. Action C will be chosen if and only if $\mathrm{E}R_i(C|G) \geq \mathrm{E}R_i(D|G)$, i.e. iff

$$\sum_{j\in N_i} q_j^t \geq |N_i|\frac{E}{N}. \qquad (47)$$

8.2. Searching for a Nash Equilibrium

In game theory, the solution concept we are dealing with most frequently is the Nash Equilibrium. In our case, we have already restricted the Good players' strategies to fictitious play, as shown in Eq. (47), so the Nash Equilibrium will be restricted in this sense.

More formally, the Nash Equilibrium in our case would be a vector $\vec{q} \in [0,1]^{|V|}$, where q_i will be the frequency with which user i plays C. The subvector $\vec{q}_G$ corresponding to the Good users will contain only 0s and 1s, according to whether Eq. (47) is false or true. For example, if Eq. (47) is false for user i, then the i^{th} element of $\vec{q}$ will be zero. The subvector $\vec{q}_B$ corresponding to the Bad users will contain values in [0,1] such that any other value in element i would not increase the payoff of Bad user i. The payoff of Bad user i when he is playing C with frequency q_i is [82]:

$$R_i(q_i|B) = (E - Nq_i)|\{j \in N_i : q_j = 1\}| \qquad (48)$$

Note that since a Bad user only has Good neighbors, and Good users only play always C or always D, the q_j ($j \in N_i$) will all be either 0 or 1.

Let us look at the case of a single Bad player in the whole network. Since no other Bad players exist, the choice of the Bad user will only affect his own payoff. We will see with what frequency he has to play C in order to maximize his payoff, in a tree topology where he is at the root.

Assume that the Bad user – labeled user 0 – has k neighbors, labeled $1, \ldots, k$. We also assume that all the Good users will start by playing C, and will only change to D if they are forced by the Bad user. Applying Eq. (47) for each neighbor, we see that each expects to see a different sum of frequencies from his own neighbors in order to keep playing C. User i expects to see a sum of frequencies that is at least $\frac{E}{N}|N_i|$. Since all of i's neighbors except user 0 are Good, they will at least start by playing C, so user i will see a sum of frequencies equal to $|N_i| - 1 + q_0$ (q_0 is the frequency with which the Bad user 0 is playing C). So, in order to make user i continue playing C, the Bad user should play C with frequency

$$q_0 \geq \frac{E}{N}|N_i| - (|N_i| - 1) = 1 - |N_i|(1 - \frac{E}{N}) \equiv t_i, \qquad (49)$$

which is decreasing with $|N_i|$, since $E < N$. We call this quantity the *threshold* t_i:

Definition 3. The *threshold* of a Good user with b Bad neighbors and g Good neighbors is the total frequency of Cs that his Bad neighbors need to play, so that he keeps playing C assuming that all his Good neighbors play C.

Without loss of generality, we assume that the Bad user's neighbors are labeled in increasing order of $t_i : t_1 < t_2 < \ldots < t_k$. By choosing $q_0 = t_j$, the Bad user guarantees that his neighbors $1, \ldots, j$ will play C, and the rest D. His payoff as a function of t_j is $R_0(t_j|B) = j(E - Nt_j)$.

The aim of the Bad user is to find the value of t_j that maximizes the payoff. For two values of q_0, say t_l and t_m with $l < m$, the payoff comparison boils down to:

$$\begin{aligned} R_0(t_l|B) > R_0(t_m|B) &\Leftrightarrow l(E - Nt_l) > m(E - Nt_m) \\ &\Leftrightarrow t_m > \frac{l}{m}t_l + (1 - \frac{l}{m})\frac{E}{N} \end{aligned} \tag{50}$$

When the Bad user chooses a value for q_0, some of his neighbors will play C and some D. The ones who play D may cause their own neighbors to start playing D, and so on. However, the Ds cannot, by propagating, influence other neighbors of the Bad user: a consequence of the tree topology.

What happens when there are multiple Bad users in a general topology? We will examine the circumstances under which the maximization of the total sum of Bad users' payoff is achieved through the local maximization of each Bad user's payoff. This local maximization is done as we have just described in Eq. (50). We call 'Uncoupled Case' the situation described by these circumstances.

Definition 4. The *tolerance* of a Good user is the largest number of his one-hop neighbors that can play D, before he starts playing D himself.

The tolerance of a Good user is a function of $\frac{E}{N}$. To compute the tolerance of user i, assume that n of his neighbors play D, and $|N_i| - n$ play C. From Eq. (47), for user i to play C the following needs to hold:

$$|N_i| - n \geq |N_i|\frac{E}{N} \Leftrightarrow n \leq |N_i|\left(1 - \frac{E}{N}\right) \tag{51}$$

The tolerance is the largest integer n for which this equation holds, i.e., $n_{\max} = \lfloor |N_i|(1 - \frac{E}{N}) \rfloor$.

All we have to do is to make sure that Good players who start playing D do not cause, recursively, 'too many' other Good users to play D so that the payoffs of other Bad users is affected. This will happen if and only if the nodes that play D because of a Bad user B_1 are separated by at least two nodes (three hops) from the nodes that play D because of any other Bad user. In other words, there needs to be a layer of nodes at least two nodes deep that have large enough tolerances so that they will not start playing D themselves. Since, for a fixed $\frac{E}{N}$, the tolerance of a user depends only on the number of his neighbors, nodes with a high degree that are connected to each other would provide the highest resistance to playing D. In graph theoretic terms, the greatest 'aggregate' tolerance is achieved, for a given number of nodes, when the nodes are connected in a clique. For further details we refer to [82].

Acknowledgments

This paper describes the work and results in my plenary address at the 2007 European Control Conference. I would like to thank Professors Tzafestas (NTUA) and Antsaklis (Notre Dame) (general chair and chair of the international program committee of the 2007 ECC) for inviting me to deliver a plenary address. The work reported here is joint work with my students Alvaro Cardenas, Tao Jiang, Svetlana Radosavac and George Theodorakopoulos. This research was supported by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, CTA on Communications and Networks, Cooperative Agreement DAAD19-01-2-0011, the U.S. Army Research Office under CIP URI grant No DAAD 19-01-1-0494, the DARPA NMS Program under Contract N66001OOC8063, the NASA RPC Program under Cooperative Agreements NCC8-235 and NAG-2844, Lockheed Martin Corporation, Hughes Network Systems and the Maryland Industrial Partnerships program (MIPS), all through the Maryland Hybrid Networks Center (HyNet), at the University of Maryland College Park.

References

1. Autonomic communication. [Online]. Available: http://www.autonomic-communication.org
2. AUVs: in space, air, water, and on the ground. IEEE Control Systems Magazine, vol. 20, pp. 15–18, December 2000
3. Aguayo D, Bicket J, Biswas S, Morriss GJR. Link-level measurements from an 802.11b mesh network. In SIGCOMM '04: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Portland, Oregon, USA, 2004, pp. 121–132
4. Anderson R. Security Engineering. John Wiley & Sons, 2001
5. Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. In Proceedings

of the 6th ACM Conference on Computer and Communications Security (CCS '99), November 1999, pp. 1–7

6. Baccelli FL, Cohen G, Olsder GJ, Quadrat JP, Synchronization and Linearity: An Algebra for Discrete Event Systems. John Wiley & Sons, 1992
7. Baras JS. Emerging wireless technologies and services: for better quality of life and work. Invited plenary address. In Proceedings of the 45th FITCE Congress, Athens, Greece, Aug. 30 – Sept. 2 2006
8. Baras JS, Cárdenas AA, Ramezani V. On-line detection of distributed attacks from space-time network flow patterns. In: Proceedings of the 23rd Army Science Conference, Orlando, FL, December 2002
9. Baras JS, Jiang T. Cooperative games, phase transitions on graphs and distributed trust in MANET. In Proceedings of the 2004 IEEE Conference on Decision and Control, Bahamas, December 2004, pp. 93–98
10. Baras JS, Jiang T. Dynamic and distributed trust for mobile ad-hoc networks. In Proceedings of the 24th Army Science Conference, Orlando, FL, November 2004
11. Baras JS, Jiang T. Cooperation, trust and games in wireless networks. In Abed EH (ed.). Advances in Control, Communication Networks, and Transportation Systems, Birkhäuser, 2005, pp. 183–202
12. Basseville M, Nikiforov I. Detection of Abrupt Changes: Theory and Application. Prentice Hall, Englewood Cliffs, NJ. 1993
13. Blaze M, Feigenbaum J, Ioannidis J, Keromytis AD. The role of trust management in distributed systems security. In Vitek J, Jensen CD (eds). Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Springer, 1999, pp. 185–210
14. Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999
15. Brémaud P. Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Texts in applied mathematics; 31. Springer-Verlag, New York. 1999
16. Buchegger S, Le Boudec J-Y. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In: Proceedings of Tenth Euromicro PDP (Parallel, Distributed and Network-based Processing), Gran Canaria, January 2002, pp. 403–410
17. Čapkun S, Buttyán L, Hubaux JP. Small worlds in security systems: an analysis of the PGP certificate graph. In Proceedings of ACM New Security Paradigms Workshop 2002, Norfolk, Virginia Beach, USA, September 2002, pp. 28–35
18. Čapkun S, Hubaux J-P, Buttyán L. Mobility helps security in ad hoc networks. In: Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2003), Annapolis, MD, June 2003
19. Cárdenas AA, Baras JS. A unified framework of information assurance for the design and analysis of security algorithms. In Proceedings of the 25th Army Science Conference, Orlando, FL, November 2006
20. Cárdenas AA, Baras JS. Evaluation of classifiers and learning rules: Considerations for security applications. In Proceedings of AAAI 06 Workshop on Evaluation Methods for Machine Learning, Boston, MA, July 2006
21. Cárdenas AA, Baras JS, Ramezani V. Distributed change detection and filtering for worms, ddos and other network attacks. In: Proceedings of the 2004 IEEE American Control Conference (ACC), pp. 1008–1013, Boston, MA, July 2004
22. Cárdenas AA, Baras JS, Seamon K. A framework for the evaluation of intrusion detection systems. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy, Digital Identifier 1081-6011/06, Oakland, CA, May 21–24, 2006
23. Cárdenas AA, Radosavac S, Baras JS. Detection and prevention of MAC layer misbehavior in ad hoc networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN 04), pp. 17–22, Washington, DC, October 25, 2004
24. Cárdenas AA, Radosavac S, Baras JS. Performance comparison of detection schemes for MAC layer misbehavior. In: Proceedings of the 26th Annual IEEE Conference on Computer Communications, (IEEE INFOCOM 2007), 2007
25. Corson S, Macker J. Mobile Ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations, RFC 2501, IETF. January 1999
26. Cowie J, Ogielski A, Premore B, Yuan Y. Internet worms and global routing instabilities. In Firoiu V, Zhang Z-L. (eds). Proceedings of SPIE – Volume 4868 Scalability and Traffic Control in IP Networks II, pp. 195–199, July 2002
27. Dellarocas C. The digitization of word-of-mouth: Promise and challenges of online reputation mechanism. Management Science, vol. 49, pp. 1407–1424, 2003
28. Di Crescenzo G, Ghosh A, Talpade R. Towards a theory of intrusion detection. In: Proceedings of the ESORICS 2005, 10th European Symposium on Research in Computer Security. Milan, Italy, September 12–14 2005. Lecture Notes in Computer Science 3679 Springer, pp. 267–286
29. Dijkstra EW. A note on two problems in connection with graphs. Numerische Mathematik, vol. 1, pp. 269–271, 1959
30. Dragalin V, Tartakovsky A, Veeravalli V. Multihypothesis Sequential probability ratio tests-part 1: Asymptotic optimality. IEEE Trans Inf Theory 1999; 45(7): 2448–2461
31. Draves R, Padhye J, Zill B. Routing in multi-radio, multi-hop wireless mesh networks. In Proceedings of the MobiCom '04, 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, 2004, pp. 114–128
32. Dutta B, Jackson MO. (eds). Networks and Groups: Models of Strategic Formation. Springer, 2003
33. Eisner J. Parameter estimation for probabilistic finite-state transducers. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, July 2002, 1–8
34. Eskin E, Arnold A, Prerau M, Portnoy L, Stolfo S. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In Barbara D, Jajodia S. (eds). Data Mining for Security Applications, Kluwer, 2002
35. Forgó F, Szép J, Szidarovsky F. Introduction to Theory of Games: Concepts, Methods, Applications. Kluwer Academic. 1999
36. Gaffney JE, Ulvila JW. Evaluation of intrusion detectors: A decision theory approach. In: Proceedings of

2001 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 2001, pp. 50–61

37. Gallegos A, Mazzag BC, Mogilner A. Two continuum models for the spreading of myxobacteria swarms. Bull Math Biol 2006; 68: 837–861
38. Gnutella. [Online]. Available: http://www.gnutella.com
39. Gu G, Fogla P, Dagon D, Lee W, Skoric B. Measuring intrusion detection capability: An information-theoretic approach. In: Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS '06), pp. 90–101, Taipei, Taiwan, March 2006
40. Guttman JD, Herzog AL. Rigorous automated network security management. The MITRE Corp., Tech. Rep., August 2003
41. Handley H, Kreibich C, Paxson V. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In: Proceedings of the 10th USENIX Security Symposium, pp. 115–131, Berkeley, CA: USENIX Association, 2001
42. Hubaux J-P, Buttyán L, Čapkun S. The quest for security in mobile ad hoc networks. In: Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001), pp. 146–155, 2001
43. IETF Mobile Ad-hoc Networks (manet) working group. [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html
44. Internet Security Systems Inc. Internet risk impact summary for June 25, 2002–September 27, 2002
45. Jadbabaie A, Lin J, Morse SA. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Autom Control 2003; 48(6): 998–1001
46. Jiang T, Baras JS. Autonomous trust establishment. In: Proceedings of the 2nd International Network Optimization Conference, Lisbon, Portugal, April 2005
47. Jiang T, Baras JS. Fundamental Tradeoffs and Constrained Coalitional Games in Autonomic Wireless Networks. In Proceedings WiOpt 2007
48. Jiang T, Baras JS. Trust evaluation in anarchy: A case study on autonomous networks. In: Proceedings of the 25th Conference on Computer Communications (IEEE Infocom 2006). 2006
49. Jiang T, Theodorakopoulos G, Baras JS. Coalition formation in manets. In Proceedings of the 25th Army Science Conference, Orlando, FL, November 2006
50. Kaufman C, Perlman R, Speciner M. Network Security – Private Communications in a Public World. Prentice Hall, 1995
51. Kindermann R, Snell JL. Markov Random Fields and Their Applications. Contemporary Mathematics. American Mathematical Society, 1980, vol. 1
52. Kruegel C, Kirda E, Mutz D, Robertson W, Vigna G. Automating mimicry attacks using static binary analysis. In Proceedings of the 2005 USENIX Security Symposium, Baltimore, MD, August 2005, pp. 161–176
53. Kruegel C, Mutz D, Robertson W, Valeur F. Bayesian event classification for intrusion detection. In Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC), December 2003, pp. 14–24
54. Kruegel C, Mutz D, Robertson W, Vigna G, Kemmerer R. Reverse engineering of network signatures. In Proceedings of the AusCERT Asia Pacific Information Technology Security Conference, Gold Coast, Australia, May 2005
55. Lee W, Stolfo SJ. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, 1998
56. Lee W, Stolfo SJ, Mok K. A data mining framework for building intrusion detection models. In Proceedings of the IEEE Symposium on Security & Privacy, Oakland, CA, USA, 1999, pp. 120–132
57. Lippmann RP, Fried DJ, Graf I, Haines JW, Kendall KR, McClung D, Weber D, et al. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In DARPA Information Survivability Conference and Exposition, vol. 2, January 2000, pp. 12–26
58. Marti S, Giuli TJ, Lai K, Baker M. Mitigating routing misbehavior in mobile ad hoc networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. Boston, Massachusetts, United States: ACM Press, 2000, pp. 255–265
59. Milgram S. The small world problem. Psychol Today 1967; 1(61): 60–67
60. Mohri M. Semiring frameworks and algorithms for shortest-distance problems. J Autom Lang Comb 2002; 7(3): 321–350
61. Moustakides GV. Quickest detection of abrupt changes for a class of random process. IEEE Trans Inf Theory 1998; 44(5): 1965–1968
62. National Science Foundation. Future Internet Network Design Program, December 2005. USA NSF FIND Program Description
63. Nishimori H. Statistical Physics of Spin Glasses and Information Processing: An Introduction. Oxford University Press, 2001
64. Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans Autom Control 2004; 49(9): 1520–1533
65. Provost F, Fawcett T. Robust classification for imprecise environments. Mach Learn 2001; 42(3): 203–231
66. Ptacek TH, Newsham TN. Insertion, evasion and denial of service: Eluding network intrusion detection. Secure Networks, Inc., Tech. Rep., January 1998
67. Radosavac S, Baras JS. Detection and performance analysis of greedy individual and colluding MAC layer attackers. In: Proceedings of the 15th IST Mobile and Wireless Summit. June 2006
68. Radosavac S, Baras JS, Moustakides GV. Impact of optimal mac layer attacks on the network layer. In: Proceedings of the SASN '06: Fourth ACM Workshop on Security of Ad hoc and Sensor Networks, Alexandria, Virginia, USA, 2006, pp. 135–146
69. Radosavac S, Cárdenas A, Baras JS, Moustakides GV. Detecting IEEE 802.11 MAC layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers. J Computer Security: Special Issue on Security of Ad Hoc and Sensor Networks 2007; 15(1): 103–128
70. Radosavac S, Moustakides GV, Baras JS, Koutsopoulos I. An analytic framework for modeling and detecting access layer misbehavior in wireless networks. To appear in ACM Transactions on Information and System Security (TISSEC)

71. Raya M, Hubaux J-P, Aad I. DOMINO: A system to detect greedy behavior in IEEE 802.11 Hotspots. In Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys2004), Boston, MA, 2004, pp. 84–97
72. Rote G. Path problems in graphs. Computing Supplementum, vol. 7, pp. 155–189, 1990. [Online]. Available: http://www.inf.fu-berlin.de/rote/Papers/postscript/Path+problems+in+graphs.ps
73. Schneier B. Applied Cryptography, 2nd ed. John Wiley, 1996
74. Shankar U, Paxson V. Active mapping: Resisting NIDS evasion without altering traffic. In: Proceedings of the 2003 IEEE Symposium on Security & Privacy, Oakland, CA, USA, 2003, pp. 44–61
75. Slikker M, den Nouweland AV. Social and Economic Networks in Cooperative Game Theory. Kluwer Academic, 2001
76. Stamp M. Information Security: Principles and Practice John Wiley & Sons, 2006
77. Staniford S, Paxson V, Weaver N. How to own the internet in your spare time. In: Proceedings of the 11th USENIX Security Symposium (Security '02), 2002
78. Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the ACM SIGCOMM '01 Conference, San Diego, CA, August 2001, pp. 149–160
79. Tan K, Killourchy K, Maxion R. Undermining an anomaly-based intrusion detection system using common exploits. In: Proceeedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002), Zurich, Switzerland, October 2002, pp. 54–73
80. Tan K, McHugh J, Killourhy K. Hiding intrusions: From the abnormal to the normal and beyond. In: Information Hiding: 5th International Workshop, Noordwijkerhout, The Netherlands, October 2002, pp. 1–17
81. Theodorakopoulos G, Baras JS. Trust evaluation in adhoc networks. In: Proceedings of the 2004 ACM workshop on Wireless security. Philadelphia, PA, USA: ACM Press, 2004, pp. 1–10
82. Theodorakopoulos G, Baras JS. A game for ad hoc network connectivity in the presence of malicious users. In: Proceedings of the IEEE Globecom 2006, San Francisco, CA, November 2006
83. Theodorakopoulos G, Baras JS. Enhancing benign user cooperation in the presence of malicious adversaries in ad hoc networks. In: Proceedings of the Second IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks (SecureComm 2006), Baltimore, MD, August 2006
84. Theodorakopoulos G, Baras JS. Linear iterations on ordered semirings for trust metric computation and attack resiliency evaluation. In: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, MTNS 2006, Kyoto, Japan, July 2006, pp. 509–514
85. Theodorakopoulos G, Baras JS. On trust models and trust evaluation metrics for ad hoc networks. IEEE J Selected Areas in Communications 2006; 24(2): 318–328
86. Theodorakopoulos G, Baras JS. Malicious users in unstructured networks. In: Proceedings of the IEEE Infocom 2007, Anchorage, AK, May 2007
87. Vigna G, Robertson W, Balzarotti D. Testing network-based intrusion detection signatures using mutant exploits. In: Proceedings of the ACM Conference on Computer and Communication Security (ACM CCS), Washington, DC, October 2004, pp. 21–30
88. Wald A. Sequential Analysis. New York: John Wiley and Sons, 1947
89. Wald A, Wolfowitz J. Optimum character of the sequential probability ratio test. Ann Math Statist vol. 19, pp. 326-339, 1948
90. Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using system calls: Alternative data models. In: Proceedings of the 1999 IEEE Symposium on Security & Privacy, Oakland, CA, USA, May 1999, pp. 133–145
91. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature 1998; 393: 440–442
92. Wierzchon S. (2002) Ising model. Eric Weisstein's World of Physics. [Online]. Available: http://scienceworld.wolfram.com/physics/IsingModel.html
93. Yang S-A, Baras JS. Modeling vulnerabilities of ad hoc routing protocols. In Proceedings of the SASN 2003, George Mason Univ., Fairfax, VA, December 1–5 2003, pp. 761–766
94. Zhang Y, Lee W, Huang Y. Intrusion detection techniques for mobile wireless networks. ACM/Kluwer Mobile Networks and Applications (MONET), vol. 9, no. 5, pp. 545–556, September 2003