J Comput Virol (2009) 5:345–355
DOI 10.1007/s11416-008-0098-9

EICAR 2008 EXTENDED VERSION

# Danger theory and collaborative filtering in MANETs

**Katherine Hoffman · Attila Ondi · Richard Ford · Marco Carvalho · Derek Brown · William H. Allen · Gerald A. Marin**

Received: 20 January 2008 / Revised: 23 June 2008 / Accepted: 8 July 2008 / Published online: 12 August 2008


**Abstract** As more organizations grasp the tremendous benefits of Mobile Ad-hoc Networks (MANETs) in tactical situations such as disaster recovery or battlefields, research has begun to focus on ways to secure such environments. Unfortunately, the very factors that make MANETs effective (fluidity, resilience, and decentralization) pose tremendous challenges for those tasked with securing such environments. Our prior work in the field led to the design of BITSI – the Biologically-Inspired Tactical Security Infrastructure. BITSI implements a simple artificial immune system based upon Danger Theory. This approach moves beyond self/non-self recognition and instead focuses on systemic damage in the form of deviation from mission parameters. In this paper, we briefly review our prior work on BITSI and our simulation environment, and then present the application of collaborative filtering techniques. Our results are encouraging, and show that collaborative filtering significantly improves classification error rate and response within the MANET environment. Finally, we explore the implications of the results for further work in the field, and describe our plans for new research.

## 1 Introduction

Computer networking has become an enabling technology for a wide variety of services. However, despite the apparent ubiquity of connectivity there still exist environments that lack fixed infrastructure to support widespread computer to computer interaction. For example, modern battlefield systems and disaster relief efforts both lack the infrastructure on which much communication is based. In such environments, connectivity is often provided by "mobile ad hoc networks" (MANETs). Such MANETs are defined in RFC2051 [5] as follows:

**Definition 1** (MANET)[5] A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communications devices)–herein simply referred to as "nodes"–which are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices, and there may be multiple hosts per router. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network.

In general, MANETs need to deal with different issues than traditional wired networks. Because there is no central infrastructure (and nodes must instead forward traffic collaboratively), each node in the network must either ask other nodes for a path to a destination on demand (reactive routing) or maintain a local view of the network topology for route calculation (proactive routing), which must be frequently updated. These can lead to issues of route disruption when nodes are accidentally or purposefully sent incorrect information about the network topology, or when critical nodes are disabled, even if temporarily. Furthermore, there exist a myriad of other security concerns in the MANET environment – for an overview, see [14] – brought about by the lack of centralized management, shifting topology, and bandwidth constrictions. As such, much work is needed if MANETs are to be used for mission-critical functions in a potentially-hostile environment.

K. Hoffman · A. Ondi · R. Ford (✉) · D. Brown · W. H. Allen · G. A. Marin
Department of Computer Sciences,
Florida Institute of Technology, 150 W. University Blvd,
Melbourne, FL 32901, USA
e-mail: rford@se.fit.edu

M. Carvalho
Institute for Human Machine Cognition,
Pensacola, FL, USA

The remainder of this paper is structured as follows. We first examine threats to MANETs and prior work in the field of security for the MANET environment. With this understanding, we then provide a short overview of our Danger Theory-inspired approach to MANET security. This framework, known as the Biologically-Inspired Tactical Security Infrastructure (BITSI), forms the basis for our experiments using reputation and collaborative filtering. The experiments are described in the next section, followed by a discussion of the results. Finally, the paper concludes with a discussion of the implication of these results to future work, and describes our plans for new research.

## 2 MANET security in general

When one considers the general structure of a MANET, it quickly becomes apparent that MANET security issues are a superset of traditional wired security problems. Thus, in addition to traditional security vulnerabilities, a MANET must also contend with the following challenges:

1. In a MANET, nodes cooperate to route traffic. Any routing algorithm must contend with nodes that may be under an attacker's control.
2. Bandwidth is locally shared and often highly-constrained in a MANET. How can this congestion be handled while simultaneously detecting nodes that are maliciously flooding the network or dropping traffic?
3. Battery life is often a concern for MANET designers, as roaming nodes often wish to act selfishly in order to conserve power. Thus, CPU cycles and wireless power management are extremely valuable commodities.
4. As the traffic observed by a node depends greatly on network topology, it is difficult for systems to learn what "good" traffic patterns look like, and what constitutes an "attack".
5. Nodes frequently enter or leave the network, causing frequent changes in network membership and contributing to localized changes in topology.
6. There is no "central authority" for network monitoring and management, as the network can become disjoint at any time.

Amongst these issues, some of the most commonly explored themes in the literature are routing attacks and selfish node behaviour. Solutions are broad, ranging from additional encryption to virtual currency and reputation systems. In terms of general security, IDS/IDP is more challenging in the MANET primarily due to the frequent changes in topology and the lack of a central authority.

Collaboration between nodes is the obvious solution, and has been examined by many other researchers. For example, Huynh, Jennings, & Shadbolt [7] examine different types of trust as a potential for improving the selection of partner agents. Similarly, Sterne et al. [14] explore the benefits of creating hierarchies within the nodes for intrusion detection.

The underlying idea is relatively simple. When a node finds another node misbehaving, it could tell other nodes about the problem, and then they could all avoid the problematic node. The trouble with reputation-based approaches is that they introduce new problems – a node could have been misidentified as harmful, and would still be shunned, or a malicious node could lie about having been hurt, potentially crippling the network. The notion of trust, as distinct from reputation was introduced to deal with this. Trust is based on most of the same information as reputation, and introduces new complications, such as whether or not to re-trust nodes that have previously been defined as malicious, and if so, when to do it, as well as what to do if malicious nodes attempt to falsely accuse good nodes of being bad.

An interesting exploration of these ideas is found in Buchegger & Le Boudec [2]. In this paper, the authors describe a system, CONFIDANT, which attempts to harden reputation systems against deliberate misinformation by looking for significant differences in reputation scores between actors. Nodes whose reputation scores for others were significantly different from the assessing node were considered less trustworthy. Several others have used similar techniques – for example, Liu & Issarny [9] and Zouridaki [16]. However, this aspect of the work is not fully explored in [2], as the experimental results are taken from a fairly simple congruency metric, as opposed to the more sophisticated dynamic trust adaptation also discussed within the work.

As can be seen, MANETs present a difficult challenge to those who would secure them. To this end, we have elected to explore biology for inspiration.

## 3 AIS and danger theory

It is our belief that a MANET security solution must be decentralized, adaptive, and resilient to both failures and attacks. Because of these requirements, a biologically-inspired approach is attractive, as natural systems often display these qualities. In particular, computer scientists have often been tantalized by the concept of building an Artificial Immune System (AIS), which can dynamically detect and adapt to new threats.

Artificial Immune Systems (AIS) have held great promise in the security field. Early work by IBM [8] and Forrest [6] focused on systems that could detect "non-self"

entities and respond to them. Despite a successful demonstration of the IBM system at the Virus Bulletin Conference in San Francisco in 1997 [8] commercially available implementations of these concepts are generally weak at best.

Part of the challenge with the AIS model is that the human immune system seems to be far more complex than simple self/non-self discrimination. For example, many non-self entities are accepted by the body (for example, parenterally-administered drugs) without provoking an immune response. Clearly, there is more at work than just discriminating between the body and "everything else".

In order to address this, Matzinger proposed that natural immune systems respond not to just self/non-self, but also detect danger [10]. When a cell dies via natural causes, well-regulated biological pathways are followed; this is called aptosis. Conversely, when a cell undergoes stress or traumatic destruction, certain danger signals are generated. This is known as cellular necrosis. While this theory is somewhat controversial among immunologists [11], the paradigm does turn out to be surprisingly helpful when constructing artificial immune systems.

AIS research including aspects of Danger Theory (DT) have begun to appear in the literature in the last few years. For example, Aikelin et al. [1] proposed the use of DT as a missing component of traditional IDS/AIS systems. This early work has sparked further exploration of such metaphors; for example, Sarafijanovic & Le Boudec [13] designed an AIS tightly linked to the biological immune system, using Danger Theory.

Danger Theory focuses on identifying and mitigating damage to the system. Note that in many cases, it is not clear if damage (for example, in the form of packet loss) is occurring simply due to the relative position between nodes (two nodes may share a poor link) or due to malicious activities. However, we note that DT is a moderator of our immune system model – only when damage is discovered does the system attempt to discern the underlying cause. The following list outlines some common attack classes and our triggers within DT:

- To protect against denial of service attacks (resource consumption), the system checks the node for resource constraints, which can include CPU load, memory utilization or network usage. Establishing thresholds (limits) on the amount of resource consumed by a single client request without triggering a reaction would not only ensure availability of service for other nodes, but can also help reserving enough resources to allow the node to further advance towards general mission objectives.
- Routing attacks are searched for when the system notes that packet loss is occurring. Note that such packet loss can occur due to environmental conditions as well as active attack. When routing errors are suspected (and packet forwarding damage is detected) the system can begin the process of determining the likely cause of problems.
- To discover the presence of worms and viruses, the system should be able to note the creation of new processes and files, plus new outbound requests. However, none of these are, at least directly, damage. Thus, from a pure DT perspective, detection will only begin if the worm/virus consumes too many resources or triggers outbound traffic that is deemed to be damaging. In our future work, our intent is to apply a policy model to system calls, associating a small level of "damage" to certain call sequences (akin to behavioural virus detection). Using this approach, our belief is that it should be possible to use a DT model for remediation of the effects of malicious code.

Of course, there are many classes of attack that would not trigger a purely-DT moderated system. For example, a user whose password had been compromised and then used maliciously would not be detected unless the attacker carried out a "damaging" action. Similarly, attacks where the damage is not immediately critical to the mission (such as data exfiltration) will not be detected using a system wholly based upon DT. As such, we argue that DT should be just one component in a larger system. This larger system is discussed below.

### 3.1 BITSI: overview

Given the security challenges of the MANET environment, our work has focused on applying theoretical concepts to real-world attacks. In particular, we have begun development of BITSI, which leverages different aspects of biological systems.

The underlying architecture of BITSI is quite straightforward. Each node of the MANET has a BITSI agent on it. This agent resides in a local trusted component at each system and monitors the behaviour of the node, as well as the traffic which is forwarded on the local network. From such a vantage point, BITSI collaboratively works to respond to different attacks.

In terms of attacks, our vision for BITSI is one of mission enablement. That is, BITSI accepts that some attacks will succeed on the network, but aims to mitigate their effects sufficiently to ensure mission continuity. This approach is different from (though synergistic with) more traditional remediation attempts, whose goal is to stop all attacks.

Remediation of attack effects is another important area of study. Softer security responses move away from binary "go/no-go" decisions toward responses which represent more of a continuum, such as rate-limiting traffic or selectively blocking connections from a particular application. By dynamically identifying and monitoring critical operations and performance requirements for specific contexts and

missions, BITSI can focus on securing the core operation of the system, as opposed to trying to address the possibly unbounded space of all possible attacks.

The challenge with such a "live and let live" approach is that it ignores the underlying sensitivity of computer data. Clearly, some information in a military environment has long term value and high criticality; others have no long term value, but are, at the short time scale, critical (an example of this might be a session key for a temporary encrypted connection). Given that this information could be extremely small in comparison to its importance (such as a 128-bit encryption key), it is very difficult to use biological techniques to prevent data exfiltration attacks, as there is no obvious biological analogy. However, this is not necessarily a fatal flaw in our approach; first, it seems unlikely that BITSI would be the only protective measure on a system; second, given the size of the problem space, a robust solution for part of the space is of value. BITSI has been designed with this in mind, and is capable of being integrated with other content-management/IDS tools.

## 4 Experimental design and goals

The work described in this paper applies collaborative filtering techniques in a Danger-Theory driven environment. It shows that while a node alone can detect and block attacking nodes, collaboration between nodes can, in many circumstances, improve detection even in the face of significantly noisy data. Furthermore, if nodes that have certain characteristics in common collaborate, and those characteristics are related to their vulnerability to attack, the results will improve still more.

These tests abstract many of the characteristics of the MANET, and were therefore carried out in a purpose-built simulator. They assume a low-mobility, tightly packed clique of nodes that are fully connected. We examine results for a subset of the nodes, which we call servers. One or more client nodes send "bad" messages representing a resource consumption attack, which cripples the receiving server for a short period, causing it to drop all subsequent messages until the bad message is processed. The server uses BITSI and the information shared by nodes to decide whether to block future messages from attacking nodes. The simulation includes a variable percentage of false positive and false negative values, which are used in this decision.

Here we introduce the notion of attributes. These attributes represent functions, qualities, services or software which each node possesses. For example, one attribute may represent the operating system used, and another may indicate the version of the Apache server the node is running. It is our contention that nodes that share many of the same attributes are more likely to be vulnerable to the same attack. Thus, we propose to use these attributes to customize the reputation information.

In order to test the effectiveness of BITSI, we examined two different scenarios. In the first scenario, we simulated a MANET network of 35 nodes, out of which 6 were assigned the role of servers that handled requests from the other nodes. One of the non-server nodes was assigned to be an attacker that only sent maliciously formed requests to the servers. Each discrete time step in the simulation was assumed to be enough for the servers to handle all legitimate requests received in that step. Three of the servers had an attribute which made them vulnerable to attack, which meant that processing an attack packet prevented servicing of all other packets within that time step. Each non-server (client) node sent 4 requests to randomly-selected servers each time step. We assumed that there was no loss of requests in the network.

Each node in the network has a BITSI client on it. This client, which is DT-inspired, classifies packets based upon their impact on the system. Thus, only packets that are serviced are evaluated by BITSI. Furthermore, we assumed that this classifier misclassifies "good" packets with probability $P_{fp}$ and "bad" packets with probability $P_{fn}$. The BITSI agent stores the classification of the last ten packets received from each node. Once this buffer is full, the oldest entry is replaced with the status of the most recent packet received. BITSI keeps such a buffer for each client encountered on the network.

Every time a packet is serviced, BITSI evaluates the contents of the buffer to determine if a particular client should be classified as an attacker and blocked for some time, $t$.

In our prior work [3], we used a SoftMax learning strategy [15] where the index of damage was calculated by the following equation:

$$\frac{e^{\eta \cdot \chi_{\text{benign}}}}{e^{\eta \cdot (\chi_{\text{benign}} + \chi_{\text{malicious}})}} > \tau \tag{1}$$

### Calculation of the Damage Index

In this equation, $e$ is Euler's number ($\sim 2.72$), $\eta$ is a learning coefficient, $\chi_{\text{benign}}$ and $\chi_{\text{malicious}}$ are the numbers of requests classified as benign and malicious, respectively, in the buffer, and $\tau$ is the decision threshold. If the inequality is true, the sending node is deemed to have caused definite damage, and some remedial action may be taken. For an examination of our previous results in this work, see [3]. In our current simulation the threshold was set to 0.5.

Once a node was identified as malicious, its "bad reputation" counter local to the server was incremented and requests from the node were blocked for an exponentially increasing number of steps based on the counter. The local "bad reputation" counter essentially served as an indicator on how many times the sender of the currently evaluated request tried to attack the server.

### 4.1 Scenario 1: results

One challenge with the work is determining how to quantify our results; that is, how can we determine how "well" BITSI is functioning? In traditional IDS/IDP systems it is relatively easy to measure the Type I and Type II error rates. However, BITSI is not a classifier *per se*, so it does not quantify traffic in this manner. Instead, BITSI will – in the most general description – attempt to preserve certain properties of the macroscopic system by reconfiguring nodes to defend themselves, sometimes at the cost of local optimality.

In similar work (for example, routing protocols) researchers have attempted to quantify "goodput" in the system; that is, the amount of legitimate requests serviced under certain conditions. However, in a real system, this is not something that can be easily done, as there is no clear cut delineation between "good" and "bad" in a system that is overcommitted in terms of resource consumption.

For the purposes of this paper, consider the following types of traffic:

- A: Legitimate traffic *sent* by nodes
- B: Legitimate traffic *serviced* by nodes
- C: Malicious traffic *sent* by attackers
- D: Malicious traffic *serviced* by vulnerable nodes
- E: Malicious traffic *serviced* by immune nodes or lost in the network

It should be noted that when a vulnerable node services a malicious attack, it becomes unable to service further traffic for the duration of the current time step. Conversely, when an immune node services malicious traffic, the node suffers no ill consequences.

Using these traffic designations, we could argue that the "optimal" strategy is where A = B – that is, where all traffic sent by "good" nodes is serviced. This approach makes sense in a simple system where there is a clear delineation between attack packets and benign traffic. However, things are significantly more complex when one considers systems that are naturally resource constrained (such as a MANET). In such a system, *any* traffic can cause some level of damage, as servicing one packet virtually guarantees that some other packet will not be serviced. In such a case, more complex metrics will need to be created. However, in this paper, as we are considering simple direct attacks, Quality of Service (QoS) is defined as $100 * \frac{B}{A}$. Thus a QoS of 100% means all "good" traffic is serviced. This metric provides a balance between penalizing the system for false positives and rewarding the system for servicing legitimate requests.

Figure 1 shows a plot of the percentage of legitimate services handled by the system at a misclassification rate ($P_{fp}$ and $P_{fn)}$ of 25%, with a threshold $\tau$ of 0.5. In this graph, the responsiveness of the system ($\eta$) was varied from 0.1 to 1.0. As can be seen, the system correctly adapts to the attackers for high values of $\eta$. However as $\eta$ decreases (corresponding to a more reactive system), the response to misclassifications begins to dominate, and the system begins to block legitimate traffic.

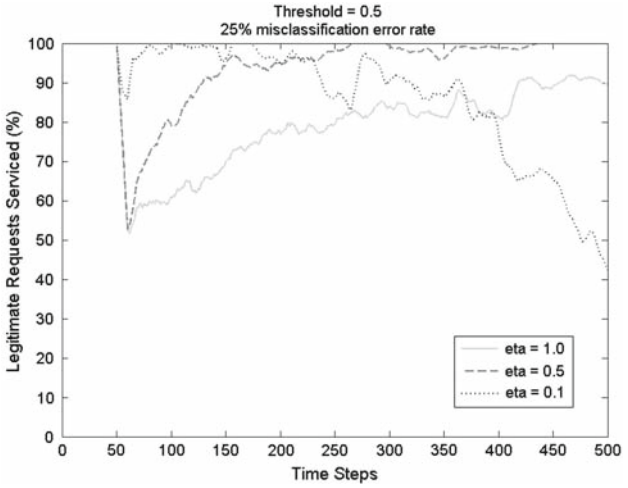


**Fig. 1** QoS for various values of Eta ($\eta$). Note how the system becomes too reactive as Eta decreases

### 4.2 Scenario 2

In the second simulation, we introduce the idea that nodes have attributes known to all other nodes in the group. These may be given in a table before the start of a mission, and could be updated periodically. We use these attributes to improve the recommendations from other nodes. For this simulation, we model 8 servers, each of which has a different set of three attributes. These servers provide service to 30 clients, of which 28 are benign. After timestep 50, the 2 attacking nodes begin to mix attack traffic in with their benign packets with probability $p$. However, a server is only vulnerable to a particular attack if it has the right attributes. Thus, two servers are vulnerable to attack A, two to attack B, two to both attacks, and two are invulnerable. Thus, at every timestep the attacker may attack one randomly chosen server, but only those with a particular attribute set will experience damage.

In this system, every time damage is detected, the server increments its local opinion regarding each client. Furthermore, after receiving an attack packet, the rest of the messages sent to that server during that timestep are dropped. A node's negative reputation gets incremented by one unit if the server identifies it as the source of damage. Otherwise, for each message received within that time step, each node gets $1/n$ units of blame, where $n$ is the number of messages processed during that timestep. The ratio between

accurate identification, where one node receives the entire blame for damage, and inaccurate identification, where all nodes sending traffic receive some portion of the blame is set by the misclassification rate, $P_m$. The false positive rate ($P_{fp}$) controls the frequency with which benign messages are considered to be harmful. These are treated just like attack messages.

For the purposes of this scenario, the blame simply increases as time goes on, rather than using the windowing scheme that was used in scenario 1.

Individual blame (primary reputation) is useful, but collaborating with others in a reputation scheme has been found to increase effectiveness [2]. In this paper, we wish to look at the effects of secondary reputation independent of primary reputation, leaving aside the issue of combining the two.

To look only at the secondary reputation, we introduce a new server, $S_{\text{new}}$, which is only vulnerable to attacker 1 and that has no prior knowledge of the behaviour of any of the clients. $S_{\text{new}}$ then determines the "global" reputation of all clients using two different techniques. First, it simply averages the opinion of all the servers in the system, as in Equation 2.

$$\mathcal{O}(S^x_{\text{new}}) = \frac{\sum_{i=1}^{n} \mathcal{O}(S^x_i)}{n} \tag{2}$$

**Calculation of Similarity (unweighted)**

Second, it calculates a weighted average based upon the Euclidean distance in attribute space it has to each other server, using Equation 3.

$$\mathcal{O}(S^x_{\text{new}}) = \frac{\sum_{i=1}^{n} \mathcal{O}(S^x_i).E(S_{\text{new}} S_i)}{n} \tag{3}$$

**Calculation of Similarity (weighted)**

where

$$E(S_{\text{new}} S_i) = 1 - \frac{\sqrt{\sum_{j=1}^{a} (S^j_{new} - S^j_i)^2}}{\sqrt{a}} \tag{4}$$

and a represents the total number of attributes for nodes in the system

**Calculation of Node Weights**

Thus, it will weight servers that have similar attributes to it more highly than those that are highly dissimilar.

To accommodate for randomness in the simulation (stemming from the selection of servers for requests), each scenario was run 50 times and the outcomes averaged.

### 4.3 Scenario 2: results

Figure 2 shows the reputation of the clients using both a simple and weighted average, from the perspective of $S_{\text{new}}$, where the underlying classifier is 100% accurate and the attacker sends an attack at every timestep. The upper graph shows the raw reputation scores. The lower graph shows the difference, in units of standard deviation, of each attacking node from the average reputation score.

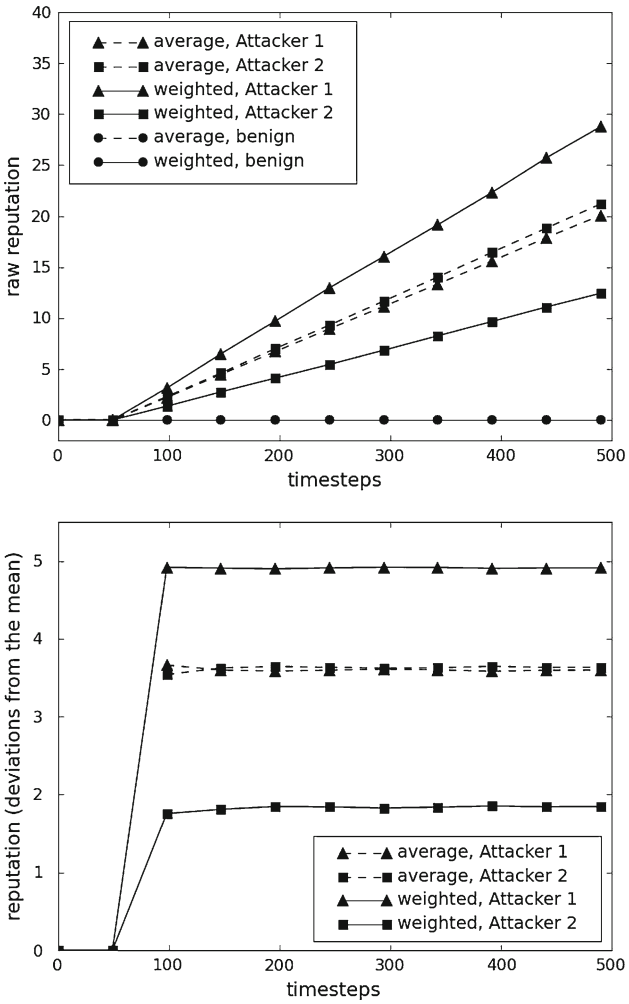


**Fig. 2** The weighted and average reputation of the client nodes as a function of time. Note how the attackers are clearly outliers from the main cluster. The same data, shown as measure of how far each node is from the mean in terms of the standard deviation

Figure 3 shows the reputation for two different false positive and misclassification rates in terms of deviations from the sample mean. In this graph – and all subsequent ones - there is a 1 in 5 chance that a particular packet sent by the attacker is an attack.

The graph on the upper side of this figure was created using a 5% false positive rate, and a 20% misclassification rate. The graph on the lower was created using a 20% false positive rate and an 80% misclassification rate. In both cases, $S_{\text{new}}$ rates attacker 1 as a significant outlier from the mean. Similarly, attacker 2, which is not capable of causing damage to $S_{\text{new}}$ lies one standard deviation from the mean, and may be treated as benign by $S_{\text{new}}$.

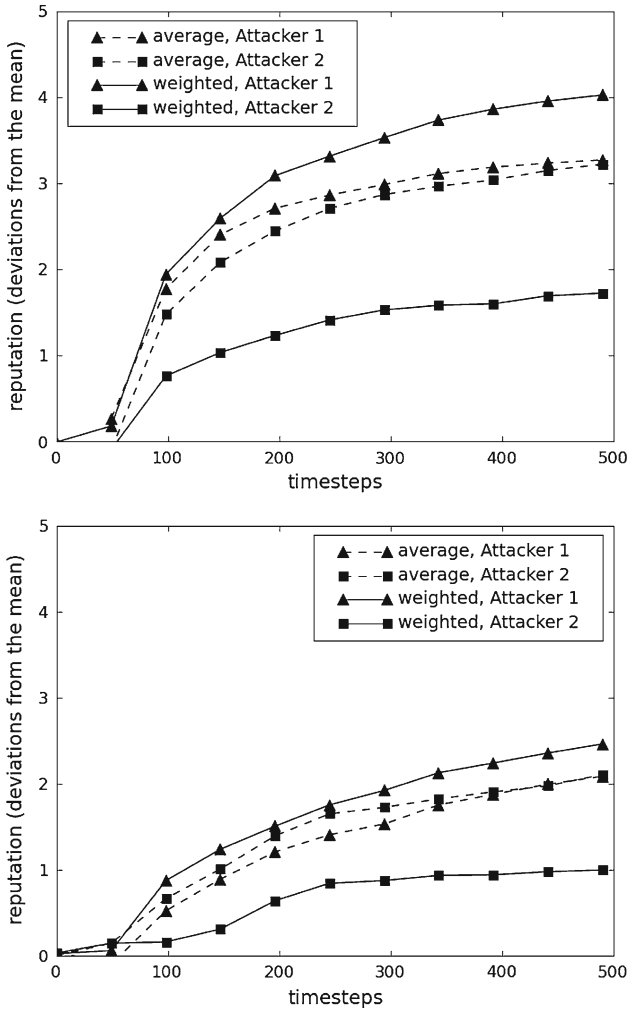


**Fig. 3** The distance from the mean of the attacking nodes for weighted and simple averaging. In the upper graph, false positives are 5% and the 20% and the misclassification rate (that is, the occurrences where BITSI detects damage but is unable to determine with certainty which traffic flow caused it) is 80%

### 4.4 Scenario 3

Similarity weighting as shown in scenario 2 allows a server to pay more attention to the opinions of nodes that are similar to it. When there are more attributes or more values for each attribute, the advantage is reduced due to more noise in the system. To counteract this, the attributes should be weighted by importance. This will not always be possible, but if a server perceives damage, and suspects a certain node, it can poll others to help decide if the node is truly dangerous. If the damage is a buffer overrun, for example, having the vulnerability in question on the host, and sharing the same Operating system is clearly more important than the presence of an unrelated scripting language on the host. Thus, the weight (importance) given to each attribute should be varied with respect to type of damage experienced.

In this scenario, $S_{\text{new}}$ perceives an attack and requests reputations from the other nodes to help determine whether or not to block the attacker. BITSI determines that there is a probability, $p$, that the attack was related to attribute $a$ – for example, its database server. It then weights attribute $a$ more heavily when calculating its similarity to other nodes. In the results below, we weighted every attribute by 1, except the relevant attribute which had a weight ($W_a$) of 5. Future work will be needed to determine appropriate attribute weights; in this work, the ratio of 5 to 1 is chosen arbitrarily.

$$E(S^x_{\text{new}}) = 1 - \frac{\sqrt{\sum_{i=1}^{a}((S^{\text{new}}_a - S^x_a).W_a)^2}}{\sqrt{\sum(W)^2}} \tag{5}$$

**Calculation of Weights (weighted by attribute)**

Note that in this equation, all distances are normalized on a scale from 0 to 1. A distance of 0 would indicate that node being compared is identical to the current node (and therefore contributes its full weight to the calculation). Conversely, a distance of 1 means a node is completely unlike the current node, and will contribute nothing to the calculation.

In this case, the opinions of servers that have the same database server are more heavily weighted than other servers. Again, each was run 50 times and averaged.

### 4.5 Scenario 3: results

Figure 4 shows a situation similar to that of scenario 2 in which misclassification is at 50% and there are 10 servers rather than 8. In (a) and (b), half of the attributes are the same as $S_{\text{new}}$. This corresponds to a case where for each attribute there are two equally likely choices. The thickest line shows the results for attribute weighting, the narrower line shows similarity-based averages and the dashed line shows simple averaging. In (a) there are three attributes, and in (b) there are 25. In the first case, similarity weighting and attribute weighting give approximately equivalent results, and both are able to distinguish the node that is dangerous from one that is less likely to be harmful. In (b), with more attributes, the similarity weighting is overwhelmed by the 24 random attributes that have no bearing on the attack, so its results are similar to those obtained using average reputation. All reputation algorithms examined can distinguish the attack nodes from the non-attack nodes, but in the attribute weighted case $S_{new}$can tell that attacker 1 is more likely to be a threat to it than attacker 2.

Figure 4(c) and (d) have the same parameters as (a) and (b), except that each attribute has only a 20% chance of being similar to $S_{\text{new}}$. This represents a situation when there are *five* equally likely choices (or the equivalent with some choices being less likely) for each attribute. Thus, there are fewer nodes that are like $S_{\text{new}}$ and so averaging will tend to give results not suited to $S_{\text{new}}$. In (c), with only three

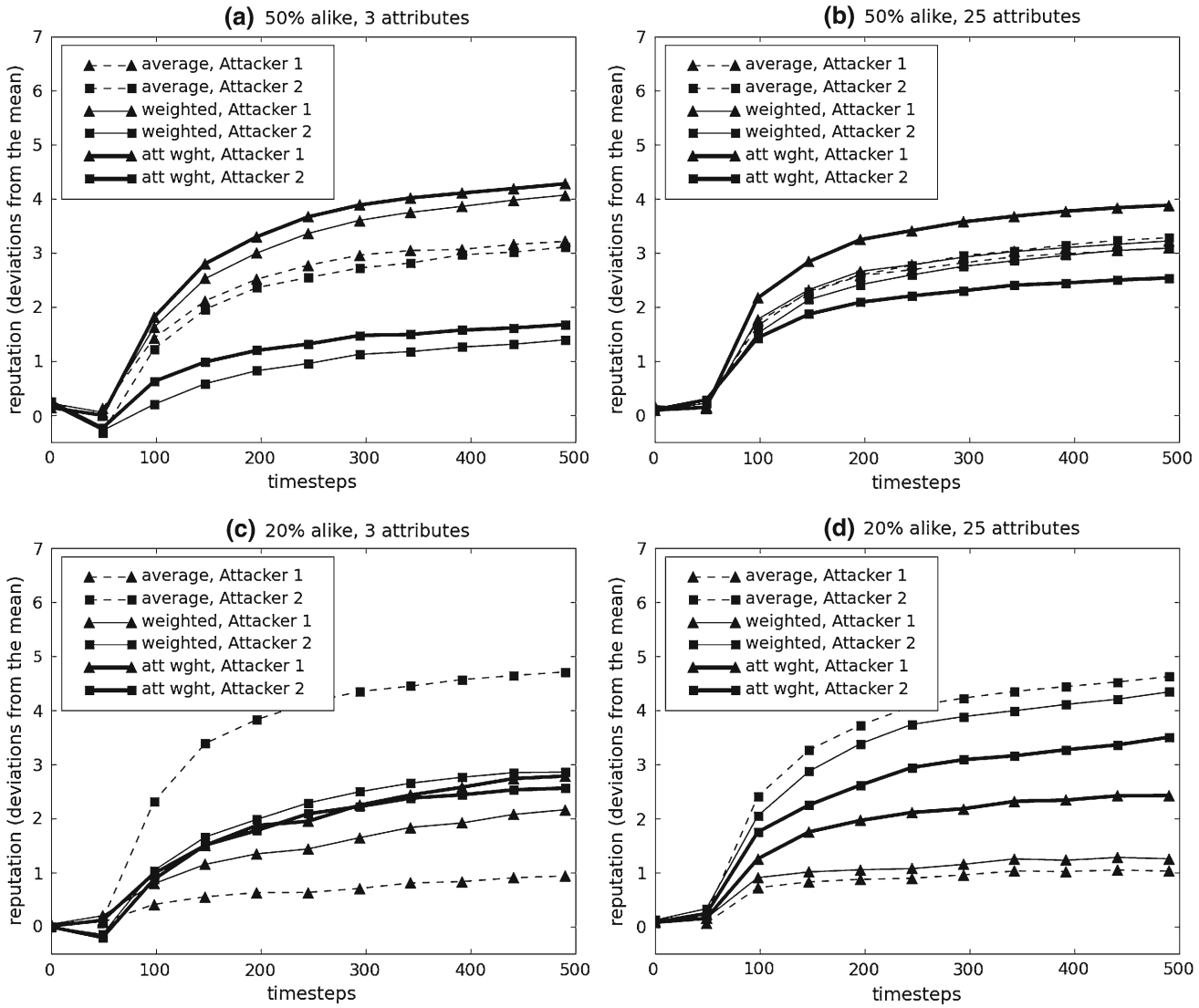


**Fig. 4** The distance from the mean of the attacking nodes for attribute weighting, similarity weighting and simple averaging

attributes, the node *not* harmful to $S_{\text{new}}$ has the worst reputation under the simple average, while for weighted averaging the two attacking nodes have about equal reputation. In (d) we show the situation with 25 attributes. In this case, attribute weighting most accurately represents the relative danger posed by each attacker.

## 5 Discussion

As illustrated in Fig. 1, the problem with a simple local reputation system is that the measurement of "badness" is not relative – that is, when the system becomes too responsive to perceived attacks, the system has no external measure of badness for comparison. Thus, as $t$ approaches infinity, all nodes are blocked. However, such fixed level approaches ignore one of the fundamental properties of the system: each node's reputation is not static, but can be compared to that of their peers. Thus, we offer two different approaches with results shown in Figs. 2, 3 and 4.

First, in Fig. 2, we show how a system that has a perfect classifier functions. Here, only the attacking nodes acquire bad reputation from its peers. Given a perfect classifier, there is obviously a trivial solution to the problem of detecting attackers. Despite this, the graphs in Fig. 2 are worthwhile studying, as they tell us something important about the system's macroscopic properties. Note how the collaborative filtering approach forces the most dangerous client (from the perspective of $S_{\text{new}}$) to have the highest negative reputation. The other attacker still has a reputation higher than the mean. This is reasonable, as $S_{\text{new}}$) is influenced primarily by the opinion of the servers most like it. Conversely, in the simple

**Table 1** Summary of the end data points in Figs. 2, 3 and 4

| Deviations from the mean | 50% Alike, 3 Attributes | | 50% Misclassification, 5% False positives | | | |
|---|---|---|---|---|---|---|
| | 20% Misclassify. | 80% Misclassify | 50% Alike | | 20% Alike | |
| | | | Attributes | | | |
| | 5% False pos. | 20% False pos. | 3 | 25 | 3 | 25 |
| Average | | | | | | |
| Node 1 | 3.29 | 2.09 | 3.21 | 3.11 | 0.95 | 1.01 |
| Node 2 | 3.24 | 2.47 | 3.11 | 3.28 | 4.72 | 4.63 |
| Similarity weighted | | | | | | |
| Node 1 | 4.04 | 2.11 | 4.06 | 3.23 | 2.18 | 1.25 |
| Node 2 | 1.73 | 0.99 | 1.41 | 3.09 | 2.89 | 4.35 |
| Attribute weighted | | | | | | |
| Node 1 | _ | _ | 4.28 | 3.89 | 2.81 | 2.43 |
| Node 2 | _ | _ | 1.69 | 2.53 | 2.58 | 3.51 |

average, both attackers are closer to the mean, and would be treated identically by $S_{\text{new}}$).

Figure 3 illustrates the real benefit of our approach. Despite the fact that attackers only attack 20% (on average) of the time, and the classifier is very unreliable (5% and 80% error rate, respectively) the node that $S_{\text{new}}$ is vulnerable to is clearly an outlier. The work described above is very promising, but requires work in several areas. In particular, we should consider the actual knowledge of the network by any node and the challenge of deliberate miscommunication by attacker nodes.

Figure 4 shows that when under attack, a node can get more information by weighting relevant attributes. This makes intuitive sense, as the node's "view" of the universe is most influenced by those nodes likely to have a similar experience.

## 6 Related work

While our prior research [3], showed promise, it was sensitive to classification error rate – as the classifier became more unreliable, the performance of the overall system declined. Furthermore, each node had to experience damage first-hand to adjust its opinion of remote nodes. Thus, in this work, our goal was to allow nodes to learn from each other's experience, by creating a reputation system.

Several researchers have tackled the reputation problem in MANETs, but in each case, there are significant differences between their approach and ours. For example, there are systems that essentially apply equal weight to each opinion (see, for example [12]). This can make sense if all players are trusted, and if the systems use equivalent methods for intrusion detection. However, in a Danger Theory-inspired system, differences in hosts' vulnerabilities change their view of the system.

Another interesting approach is to consider how much another node's view of the world is similar to your own [2]. Thus, if the opinions of Node B are very similar to those of Node A in general, Node A will tend to provide higher weight to its opinions. This approach is interesting for a DT-inspired system, as nodes with similar vulnerabilities may have fairly similar views of global reputation. In the long term, it would be interesting to implement this technique using BITSI and compare results.

Our implementation is different from these previous systems as it focuses on differences in the nodes themselves – that is, the greater the similarity between two nodes' configuration, the larger the influence each has on the other's reputation. As we have demonstrated in this work, a similarity metric based on the attributes of the nodes provides a better signal-to-noise ratio for defenders, and outperforms a simple average.

## 7 Future work

The assumption of global knowledge is clearly inappropriate for the MANET environment. Even when the network is fully-connected, it is not possible to make decisions based upon exact knowledge of the current state of the system. In the real world, however, the situation is significantly worse, as the network is unlikely to be fully connected. Thus, it is imperative that BITSI can function with only partial knowledge.

Fortunately, the fragmented nature of the MANET is not an insurmountable problem. As connectivity is required between two nodes for an attack to take place, the current connected system can be treated as the global space. In addition, it is not clear that a global view of the network helps. For example, the local reputation of a misbehaving node in

an isolated cluster is of more importance than the reputation more widely among nodes that cannot have been affected by it. Our sense is that local machines could identify and block damaged/malicious systems, and provide warnings to new nodes when the network topology changes.

The challenge of targeted attacks is a difficult one, though it is fortunately not without precedent in the literature. In any reputation-based system, if the number of attackers is large, it might be possible to skew results, if attackers collaborate. In addition, any system has to be careful to avoid strong positive feedback, where a series of false positives can cause a cascade of negative reports about a node.

In both these instances, one attractive approach is to conserve the reservoir of negative reputation and have nodes "own" the negative reputation they distribute. In [4], a system is proposed where any node may revoke another's network access… by voluntarily giving up its own. The work is interesting, as it provides strong defence to Byzantine attacks – an attacker can only use the system to remove one defender at best. Our intuition is that a modified version of this system, where one owns the bad reputation one distributes, could also be effective; this is left as an avenue for further research.

The most general way to consider our system is that the decision to block and the duration of a block are function of local knowledge and group knowledge. The primary difference between a global reputation system and collaborative filtering is that a collaborative approach weights the opinion of neighbours based upon their similarity to us. In future work, we foresee two primary research areas here: the exact nature of the classifier/blocking function, and the correct way to handle similarity metrics.

Determining the most effective form of the functions used will require an empirical approach. Furthermore, it seems likely that the optimal strategy will depend on the underlying values of $P_{fp}$ and $P_{fn}$ and the attack strategy implemented. Thus, our intent is to explore the solution space and determine if there is a set of functions that performs acceptably under a wide range of circumstances.

In terms of determining "likeness" to neighbours, there are a significant number of research avenues. For example, the metric for similarity may depend greatly on the type of attack encountered. If the attack under consideration is on a web server, for example, over port 443 (HTTPS), it makes sense to weight other web servers that support HTTPS far more highly than others. Thus, determining similarity depends on *context* (what attack is being considered right now) and attributes (what is the machine under consideration). If we were to naively assign attributes to each machine, it is possible to calculate the Euclidean distance between their attributes; however, this ignores the context issue outlined above. Once again, determining the optimum distance metric to use is a matter of considerable interest, and is an area of future research.

## 8 Conclusions

In this paper, we have outlined a Danger Theory based Artificial Immune System for the MANET environment. In particular, we have shown how such an approach can have quite desirable properties macroscopically, by focusing on high-level needs. We then showed how a simple reputation system can be improved in this environment by considering the experiences of similar systems.

Overall, the results provided are very encouraging. By focusing on high-level systemic properties, the resilience of the system is protected, and the overall mission enabled. Furthermore, the system does not attempt to impute motive to actions; instead, when using Danger Theory, the *results* of any action are analysed. Finally, the system can operate synergistically with existing techniques (such as signature-based IDS solutions) provided some estimate of the false positive error rate is known.

There remains a large amount of work to conduct before BITSI is ready for deployment. The two primary areas of concern are the lack of global knowledge and dealing with attackers who attempt to fool the system. Our hope is to continue to expand the models underpinning BITSI to deal with these circumstances.

**Acknowledgments** This work is part of a multi-institutional effort, under sponsorship of the Army Research Laboratory via Cooperative Agreement No. W911NF-07-2-0022, CFDA No. 12.630.

## References

1. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger theory: the link between AIS and IDS? In: Timmis, J. Bentley, P. Hart, E. (eds.) 2nd International Conference in Artificial Immune Systems (ICARIS 2003), pp. 147–155. Springer, Berlin (2003)
2. Buchegger, S., Le Boudec, J.Y.: The Effect of rumor spreading in reputation systems for mobile ad-hoc networks. In: WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France (2003)
3. Carvalho, M., Ford, R.A., Allen, W.H., Marin, G.: Securing MANETs with BITSI: danger theory and mission continuity, Accepted to SPIE 2008, Orlando (2008)
4. Clulow, J., Moore, T.: Suicide for the common good: a new strategy for credential revocation in self-organizing systems, In: SIGOPS Oper. Syst. Rev. **40**(3),18-21,ISSN 0163–5980 (2006)
5. Corson, S., Macker, J.: Mobile Ad hoc networking (MANET): routing protocol performance issues and evaluation considerations, IETF RFC2501 (1999)
6. Forrest, S.A., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for unix processes. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE Computer Society Press, Los Alamitos. ISBN 0-8186-7417-2 (1996)
7. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. J. Auton. Agent. and Multi-Agent Syst. 13 (2). pp. 119-154. ISSN 1387-2532 (2006)
8. Kephart, J.O., Sorkin, G., Swimmer, M., White, S.R.: Blueprint for a computer immune system. In: Proceedings of the

International Virus Bulletin Conference, Virus Bulletin PLC, San Francisco (1997)
9. Liu, J., Issarny, V.: Enhanced reputation mechanism for mobile ad hoc networks. Lecture Notes in Computer Science, vol. 2995/2004, pp. 48-62. Springer, Berlin, Book Trust Management, doi:10.1007/696545, ISBN 978-3-540-21312-3, ISSN 0302-9743 (2004)
10. Matzinger, P.: Tolerance, danger and the extended family. Annu. Rev. Immunol. **12**:991–1045 (1994)
11. Matzinger, P.: Essay 1: the danger model in its historical context. Scand. J. Immunol. **54**(1/2):4–9 (2001)
12. Repantis, T., Kalogeraki, V.: Decentralized trust management for ad-hoc peer-to-peer networks. In: MPAC '06: Proceedings of the 4th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006), p. 6. ACM Press, Melbourne (2006)
13. Sarafijanovic, S., Le Boudec, J.: An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. In: Neural Networks, IEEE Transactions on, vol. 16, no 5, pp. 1076–1087, ISSN 1045–9227 (2005)
14. Sterne, D., Balasubramanyam, P., Carman, D., Wilson, B., Talpade, R., Ko, C., Balapari, R., Tseng, C.-Y., Bowen, T., Levitt, K., Rowe, J.: (2005) A general cooperative intrusion detection architecture for MANETs. In: Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA'05), pp. 57–70, IEEE Computer Society, Washington
15. Sutton, R., Barto, A.: Reinforcement learning: an introduction. MIT Press, Cambridge. ISBN-13:978-0-262-19398-6 (1998)
16. Zouridaki, C., Mark, B.L., Hejmo, M., Thomas, R.K.: (2006) Robust cooperative trust establishment for MANETs. In: SASN '06: Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 23-34, ACM Press, Alexandria