Adaptable Misbehavior Detection and Isolation in Wireless Ad Hoc Networks Using Policies

Oscar F. Gonzalez Duque[1], Antonis M. Hadjiantonis[1], George Pavlou[2], Michael Howarth[1]
[1] Centre for Communications Systems Research (CCSR), University of Surrey, Guildford, UK.
[2] Networks and Services Research Lab, University College London, UK

***Abstract*—Wireless ad hoc networks provide the communications platform for new technologies and applications, such as vehicular ad hoc networks or wireless mesh networks. However, their multihop wireless nature makes them inherently unreliable and vulnerable, since their overall performance depends on the cooperative packet forwarding behavior of each individual node. In this paper we present a role-based approach that uses a distributed management overlay and gathers information about the packet forwarding activities of each node in the network. Using policies to control an adaptive algorithmic method that monitors the individual behavior of each node, we show that it is possible to detect, accuse and punish misbehaving nodes with a high degree of confidence. Our evaluation results demonstrate that after the successful detection of misbehaving nodes, their punishment through network isolation can significantly improve network performance in terms of packet delivery and throughput.**

***Index Terms*— wireless ad hoc networks, misbehavior detection, policy-based management, self-protection.**

I. Introduction

Wireless Ad Hoc Networks consist of end-user devices capable of multihop communication, optionally supported by limited infrastructure. This definition attempts to approach wireless and mobile ad hoc networking as a paradigm rather than as a specific technology [1, 2]. As MANETs become integrated with today's networks there is an increasing need to make them reliable. In such multihop environments where users rely on their peers for the forwarding of packets towards destinations, it is essential that the packet forwarding functionality is not compromised. For example, communication between peers can be affected by the presence of malicious nodes that decide to misbehave by dropping a subset of packets in order to hinder the overall network performance while avoiding security measures in place. Network layer misbehavior can be divided into two categories [3]: routing misbehavior (failure to behave in accordance with a routing protocol) and packet forwarding misbehavior (failure to correctly forward packets on behalf of other network peers). Our proposed approach focuses on the protection of the data forwarding functionality in multihop wireless networks. Although this approach can be integrated with routing protection schemes, we consider routing misbehavior out of the scope of this work.

Our protection scheme consists of two modules in each node: one that works in the network layer and another that works in the management plane. The network layer module consists of an algorithm that performs three tasks: 1) collects and aggregates behavior metrics on the active nodes in the network, 2) detects misbehaving nodes that maliciously drop packets above a configurable limit, and 3) accuses and punishes nodes that are persistently detected to be misbehaving. The management plane module uses policies to control and adjust the network layer module. The interworking of both modules drives network adaptation based on current conditions and the objectives of high-level entities, such as a network administrator or manager.

The rest of this paper is organized as follows. Section II describes related work in the area of multihop wireless network security and policy-based management. Section III introduces and presents our proposed protection scheme, while Section IV presents a case study and the evaluation of our approach. Finally, the paper is concluded in Section V along with directions for future work.

II. Background and Related Work

An evolving approach towards MANETs [1, 2] is evident in their definition by the IETF, which has taken a pragmatic perspective regarding research in wireless ad hoc networking. In 1999, IETF's MANET WG defined a MANET as an autonomous system of mobile nodes (RFC2501). In 2007, the newly formed IETF AUTOCONF WG (Ad-Hoc Network Autoconfiguration) defined a MANET as a loosely connected domain of routers. In addition, market momentum [2, 4] gives a renewed view of ad hoc networking. This is indicative of the abandonment of MANETs' isolation and their need to coexist and integrate with today's networks. A wealth of work can be found in the literature, investigating both misbehavior protection mechanisms at the network layer and policy-based management. However, to the best of our knowledge no previous work addresses misbehavior protection with the use of management level policies.

Approaches have been proposed that provide security in existing ad hoc routing protocols through the enhancement and removal of some of their features. Examples of such protocols are the secure efficient distance vector (SEAD) routing [5] which is based on the destination sequenced distance vector (DSDV) [6], and the secure on-demand distance vector (SAODV) routing protocol [7] based on AODV [8]. Extending the dynamic source routing (DSR) protocol to provide it with

This work was supported in part by the EU EMANICS Network of Excellence on the Management of Next Generation Networks IST-026854.

978-1-4244-3487-9/09/$25.00 ©

security mechanisms is the secure on-demand routing protocol for ad hoc networks (Ariadne) [9] and CONFIDANT (Cooperation of Nodes: Fairness In Dynamic Ad-hoc NeTworks) [10]. These approaches are complementary to our packet forwarding protection scheme since they secure the path discovery and establishment functionality of routing protocols.

There has been work that aims to provide reliable network connectivity by detecting packet forwarding misbehavior. WATCHERS (Watching for Anomalies in Transit Conservation: a Heuristic for Ensuring Routing Security) [11] is a protocol designed to detect disruptive routers in fixed networks by making use of the principle of conservation of flow in a similar way to our approach. However, WATCHERS is designed to work in fixed networks and is not applicable to MANETs.

Traffic transmission patterns have been studied [12] as a means to detect packet forwarding misbehavior using medium access control (MAC) layer techniques that preserve the statistical packet forwarding regularity from hop to hop. SCAN (self-organized network layer security in mobile ad hoc networks) [3] focuses on securing packet delivery. SCAN uses AODV [8] but argues that the same ideas can be adapted to other routing protocols. The authors of [13] propose a system based on DSR composed of two modules that reside in all network nodes: a watchdog which identifies misbehaving nodes and a pathrater which helps DSR to avoid such nodes.

As observed, the above approaches rely on predefined parameters and hard-wired logic to execute the required protection tasks. This results in a lack of flexibility and the failure to integrate with and adapt to the network management objectives. In addition, they need to be configured by a specialist familiar with the underlying algorithms. The solutions to these issues of network management have been envisioned in the policy-based management (PBM) paradigm. PBM simplifies the complex management tasks of large scale systems, since high-level policies monitor the network and automatically enforce appropriate actions in the system [14]. In general, policies are defined as Event-Condition-Action (ECA) clauses, where on event(s) E, if condition(s) C is(are) true, then action(s) A is(are) executed. PBM approaches for wireless networks have been proposed in [15, 16, 17]. At the same time, industry envisions autonomic computing as dynamically managed by business rules and policies [18], fuelling further interest in policy-based solutions. The basic components of a PBM system are the Policy Repository (PR), the Policy Management Tool (PMT), the Policy Decision Point (PDP) and the Policy Enforcement Points (PEP) [19]. The PR encapsulates the management logic to be enforced on networked entities. It is the central point where policies are stored by managers using a PMT and can be subsequently retrieved by PDPs. Relevant policies can be retrieved and interpreted by a PDP, which in turn provisions any decisions or actions to its controlled PEPs. Motivated by the capability of PBM systems for adaptability, our work employs a policy-based framework for MANETs [15, 16] as the management overlay where policies are defined and the network is algorithmically organized [20].

III. Adaptable Misbehavior Detection and Isolation

Our mechanism uses an adaptable method to detect packet forwarding misbehavior based on the principle of flow conservation [11] and the use of policy-based management (PBM) [14]. Such adaptability allows the system to judge the behavior of nodes and decide whether they should, or not, be accused of misbehavior and penalized according to current network management policies. Our approach is deployed over a role-based wireless network, organized in a hybrid tiered manner [15]. Nodes are assigned a role that defines the tasks they are responsible for as well as the policies that apply to them. For example, depending on their role, nodes may hold behavior information about their neighbors, a localized network section or the entire network.

Our approach differs from existent schemes, such as our previous work [21], since it does not rely on promiscuous listening to determine either the nodes that are not forwarding packets or the active neighborhood of a node. All relevant calculations and subsequent decisions are based on metrics directly acquired by nodes actively sending and receiving packets. Likewise, the problem posed by nodes that constantly change their geographical position in a clever manner (without going back to previously visited areas) to avoid the security measures in place is solved in this new scheme because nodes making the accusation decision have a holistic network misbehavior view. Also, to the best of our knowledge, our proposal is the first attempting to connect misbehavior detection and accusation with the use of policies at the management plane. By allowing policies to manipulate key features of our algorithm, such as the misbehavior detection threshold and the maximum expected percentage of false accusations, the network manager or administrator can fulfill the requirements stipulated by high level management goals, e.g. the desired security level.

A. Detection Phase

Our approach can be divided into two main phases: gathering of behavior information for misbehavior detection, and accusation with penalty enforcement. The former involves collecting and aggregating behavior information in the low levels of the hybrid tiered network and taking it to the top level for analysis in order to detect misbehaving nodes. The second phase on the other hand starts at the top level by deciding which nodes to accuse of misbehavior and how to penalize them, and continues with the enforcement of the respective penalties by the bottom tiers.

The proposed detection procedure is based on the clustered organization of the underlying wireless ad hoc network.

Clustering is an effective method to reduce traffic overhead in wireless networks, both for routing (e.g. the selection of multipoint relay nodes in OLSR [22]) and management purposes [15]. By selecting cluster heads and forming clusters, scalability can be increased and locality is preserved. Therefore we have adopted an existing policy-based framework to facilitate the role-based detection phase. By exploiting a fully distributed algorithm for cluster creation and maintenance [20], the framework dynamically assigns a role to each device, taking into consideration its capabilities and mobility attributes [16]. Three roles are defined, namely manager node (MN), cluster head (CH) and cluster node (CN). For modularity, a MN encapsulates the functionality of a CH and in turn a CH encapsulates that of a CN. An algorithmic process is used for dynamic cluster creation by selecting the most capable nodes as cluster heads, while remaining nodes become CNs and register with their nearest CH. Details and evaluation of the algorithm in MANETs can be found in [16]. A CH uses a PDP to locally manage the PEP of CNs that belong to its cluster and communicates with other CHs to exchange management information, including policies. Management policies are defined at MNs and are distributed to CHs for enforcement on CNs. For the rest of this paper, we assume a clustered wireless ad hoc network with a pre-assigned number of MNs and dynamic algorithmic selection of CHs.

Misbehavior detection in our algorithm is based on the principle of flow conservation [11]. This states that all bytes/packets that enter a node that is not their destination are expected to exit the node. In order to apply this principle in MANETs it is necessary to keep track of the number of packets successfully forwarded by each node in the network. In our approach each network node v_i, regardless of its role, maintains a table with two metrics T_{ij} and F_{ji} for every other node v_j to which v_i has either sent or received packets. T_{ij} is the number of packets that node v_i has transmitted to node v_j for v_j to forward to another node, and F_{ji} is the number of packets that node v_j has forwarded to node v_i that did not originate at v_j. Thus, if we consider U_j to be the neighborhood of node v_j excluding itself, Eq. 1 holds for well behaved nodes when $\alpha_{threshold}$ is set to an appropriate value as we shall see later. [21] gives a more formal definition and detailed explanation of the principle of flow conservation applied to MANETs.

$$\sum_{\forall i | v_i \in U_j} F_{ji} \geq (1 - \alpha_{threshold}) \sum_{\forall i | v_i \in U_j} T_{ij} \qquad (1)$$

Eq. 1 states that the number of packets forwarded by node v_j should be at least a fraction of those packets transmitted to v_j for v_j to forward. The parameter $\alpha_{threshold}$ lies between 0 and 1 and is the fraction of packets that a node is allowed not to forward without being detected as misbehaving.

Eq. 1 is applied to all nodes in the network, which requires the gathering and aggregation of the T_{ij} and F_{ji} metrics for each active node in the network. Our scheme achieves this by requiring all CNs to report their collected T_{ij} and F_{ji} metrics to their respective CHs periodically. The period that elapses between two consecutive reports depends on the current network management policies. CHs aggregate the reported metrics including their own. To anticipate node movements, aggregated cluster information is passed on from CHs to a MN. Finally, MNs exchange behavior information between them and perform a new metrics aggregation. At this point, MNs have acquired information on the overall network behavior as well as on the individual behavior of each active node in the network. This allows for the detection of misbehaving nodes by applying Eq. 1 to the collected metrics.

Selecting an appropriate misbehavior threshold $\alpha_{threshold}$ is very important to avoid false detections, i.e. a state where a well-behaved node is mistaken for a misbehaving one. Fig. 1 illustrates this concept. Simulation details are given in Section IV.B. Fig. 1 shows three curves depicting the percentage of detections as a function of increasing misbehavior threshold for three networks of 60 nodes where all nodes drop the same percentage of packets. This means that in the figure, the normal node behavior for each network is to drop 70%, 30% and 0% of the packets respectively. For these three cases, it is assumed that this average behavior is the normal behavior of a well behaved node, perhaps due to increased noise or mobility issues, and nodes are not expected to be detected as misbehaving. Therefore the misbehavior threshold $\alpha_{threshold}$ should at least be set to a value equal to the packet dropping average plus an offset which helps preventing our algorithm from falsely detecting well-behaved nodes. For example, in the case of the network where nodes drop 30% of its packets $\alpha_{threshold} \geq 30\% + 14\%$ in order to have less than 10% chance of detecting a node as misbehaving. In Section III.B we introduce a mathematical method that helps selecting an adequate offset value.

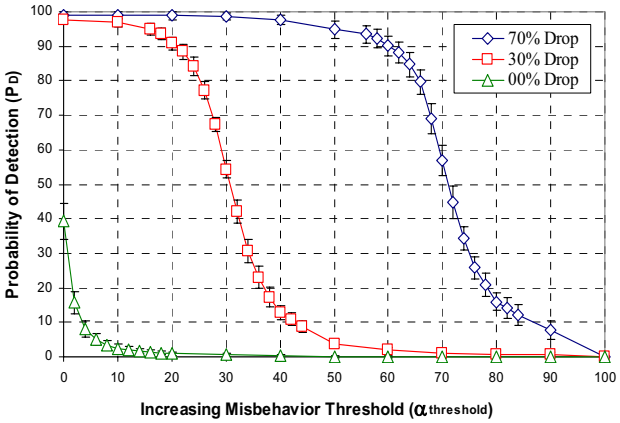

Figure 1. Probability of detection P_D as a function of the misbehavior threshold $\alpha_{threshold}$ for different levels of normal network behavior

Simulations have also been carried out for 20, 40 and 120 node networks. The curves obtained exhibit the same behavior as the 60 node network presented in this section. However, our

results show that the offset required to give a desired detection probability changes slightly between different networks.

B. Accusation Phase

As described in the previous section the network behavior information is collected, aggregated and sent to MNs for them to analyze. Thus, MNs have access not only to the overall network behavior but also to each node's individual behavior data. This makes them responsible for controlling and adjusting the procedures carried out to detect and accuse misbehaving nodes. Therefore the accusation concept developed in this section is implemented by MNs.

A single detection should not be considered sufficient to accuse a node of misbehavior since this results in a system where the probability of wrongly accusing nodes is very high. For this reason, we propose that a node in our scheme be accused of misbehavior only if in a number of behavior checks *ch* the node is detected at least *d* times as misbehaving. Since the order of those *d* detections does not matter, combinatorics and probability theory can be used to derive the following equation:

$$P_A = P_{ch,d} = \sum_{i=d}^{ch} {}_{ch}C_d \times P_D{}^{i} \times \overline{P_D}^{(ch-i)} \qquad (2)$$

where,

$P_A \rightarrow$ Probability of accusation

$P_{ch,d} \rightarrow$ Prob. of at least d detections in ch behavior checks

$P_D \rightarrow$ Probability of a detection

$\overline{P_D} = 1 - P_D \rightarrow$ Probability of a no-detection

$${}_{ch}C_d = \binom{ch}{d} = \frac{ch!}{d!(ch-d)!}, \quad ch \geq d$$

It follows that the probability of accusation P_A depends not only on the detection probability P_D but also on the number of behavior checks *ch*, which defines our sliding window (memory buffer) size, and on the minimum number of detections *d* required for a node to be accused of misbehavior. Figures 2 and 3 show how Eq. 2 varies with *ch* and *d*.

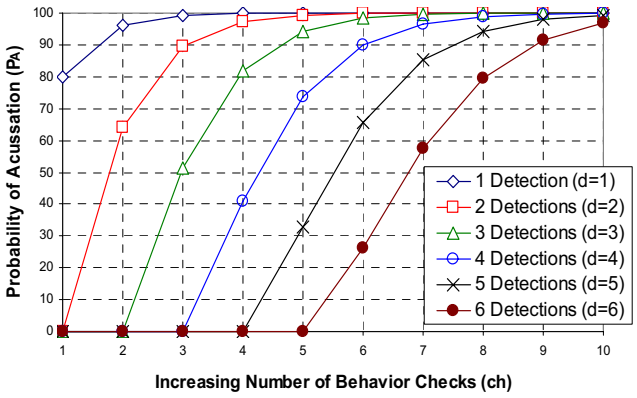

Figure 2. Probability of accusation (P_A) as a function of behavior checks (ch) for a number of required detections (d) with P_D = 80%

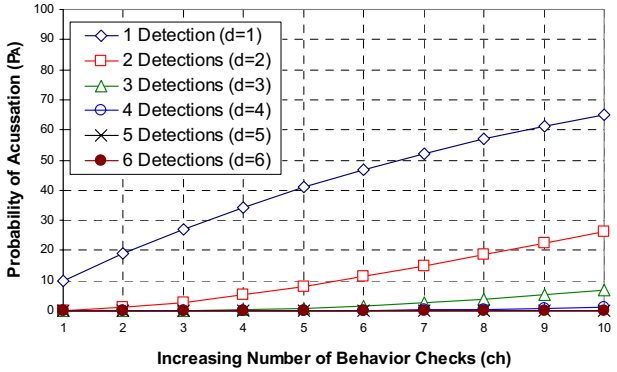

Figure 3. Probability of accusation (P_A) as a function of behavior checks (ch) for a number of required detections (d) with P_D = 10%

Fig. 2 presents a set of curves corresponding to a required minimum number of detections when the probability of a single detection is 80% (P_D = 80%). In this graph, the probability of accusation P_A increases to almost 100% in all cases as the number of behavior checks increases from 1 to 10. Fig. 3 presents the same information for probability of detection P_D = 10%. Here the maximum P_A reached is 65% and corresponds to the case when at least 1 detection out of 10 behavior checks is needed to accuse a node of misbehavior.

Figures 2 and 3 allow us to select appropriate values for the number of behavior checks *ch* and the minimum number of detections *d* needed to accuse a node. For example, assume a 60 node network consisting of two types of nodes: well-behaved nodes that do not drop packets on purpose, and misbehaving nodes that drop 70% of the packets they are supposed to forward.

Fig. 4 shows separately the probability of detection (P_D) as a function of the increasing misbehavior threshold ($\alpha_{threshold}$) for well-behaved nodes, misbehaving nodes, and their average behavior for the network in two different states. In state 1 – the initial state – 50% of nodes are well-behaved and 50% are misbehaving. In state 2 the network has detected and isolated half the misbehaving nodes, i.e. 25% of the total number of nodes in the network. We see that the state 2 curve resembles more the curve for well behaved nodes. In general the more misbehaving nodes are isolated the more the average behavior of the network resembles that of a network with well-behaved nodes only.

If in our example network, regardless of its state, we set $\alpha_{threshold}$ = 5% we see from Fig. 4 that P_D = 10% for well behaved nodes (0% drop). Then if we want to have a probability of wrongly accusing well behaved nodes no greater than 1%, from Fig. 3 we observe that setting the number of behavior checks *ch* = 4 and the minimum number of required detections *d* = 2 would not satisfy our requirement since in such case the probability of accusation P_A is about 5%. Instead if *ch* = 5 and *d* = 3 we obtain that P_A is less than 1% (actually from equation 2 we know P_A = 0.856%). Furthermore, with $\alpha_{threshold}$ = 5% the probability of detecting misbehaving nodes

is $P_D \approx 98\%$ (from Fig. 4, 70% drop) and by using equation 2 (with $ch = 5$ and $d = 3$) we obtain that the probability of accusing misbehaving nodes in our example network is virtually 100%. This proof of concept example verifies an important property of our protection scheme, i.e. the property of exhibiting a low probability of wrongly accusing well behaved nodes while maintaining a high likelihood of accusing misbehaving ones.

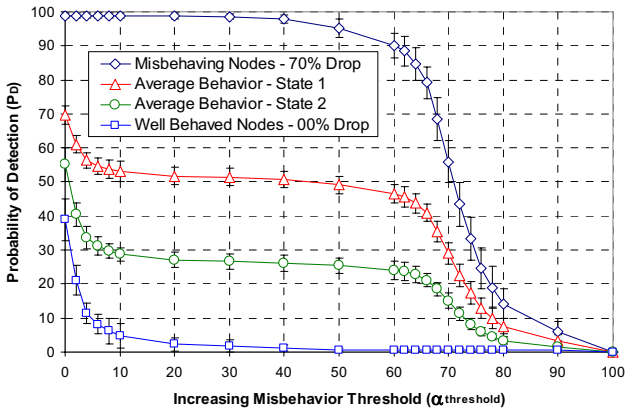

Figure 4. Probability of detections P_D as a function of the misbehavior threshold $\alpha_{threshold}$ for a 60 node network in two different states

Our algorithm's parameters can be manipulated through low-level policies in order to adapt the system to the current network conditions and at the same time fulfill the goals specified by high-level entities. On the other hand, our scheme also provides the management plane with information related to the current performance of the network. For example, event-driven policies could be triggered when the overall misbehavior of the network exceeds a pre-established limit. These policies would execute a set of procedures and parameter configurations in order to lead the overall system to a new desired state, as explained in the following section.

C. Adaptability and management policies

The adaptability of detection and accusation processes is achieved in two ways. First, a new method is proposed for the calculation of the detection threshold. This configurable method uses a weighted algorithm of recent metrics and inherently adapts to the dynamic forwarding behavior of participant nodes. The second aspect of adaptability is achieved with the use of policies, based on the aforementioned role-based network organization. Special policies are introduced to manage the proposed protection scheme, taking into account management objectives and the changing network conditions.

Policies can express the high-level goals of a managing entity and can be interpreted into low-level policies that dynamically control the operation of participating wireless devices. Although policy refinement is currently out of the scope of our work, we propose a simplified translation of high-level policies (e.g. the level of detection rigidness) and variables (e.g. the probability of wrong accusation) to low-level algorithm configuration (e.g. values for *ch* and *d*). This functionality enables an entity to predefine the expected behavior and performance of the network through policies.

We have adopted the established ECA policy notation, where a policy can be expressed as a statement in the form of: *<on Event> if <Condition(s)>then<Action(s)>*. In spite of its simplicity, this notation is quite effective because it provides the building blocks for complex management logic. This is achieved by creating groups of policies that can be assigned to management or organizational roles. For example, different policies can apply to devices in the CH and CN role. The use of events and conditional expressions in policies provides a dynamic element to the management system, making it able to adapt to network conditions. E.g. policies can be triggered if the node density in an area has become lower than a configurable threshold. A sample series of policy types have been defined, as seen in Fig.5.

We classify management policies in two sets, the *protection scheme* set and the *network deployment* set. Policies in the protection scheme set are organized in the *security requirements* and *penalization* policy groups. The network deployment policy set includes an *exceptions* policy group. Each policy group contains a number of policy rules that express the low-level policies to be enforced on network devices. The modeling of policies in sets, groups and rules follows the recommended practices of IETF, as described in RFC3460 [23]. The use of multiple policies and dynamic conditions to affect the values of the same managed objects, introduces the risk of policy conflicts and consequently the need to address conflict detection and resolution (CDR) in our future work.

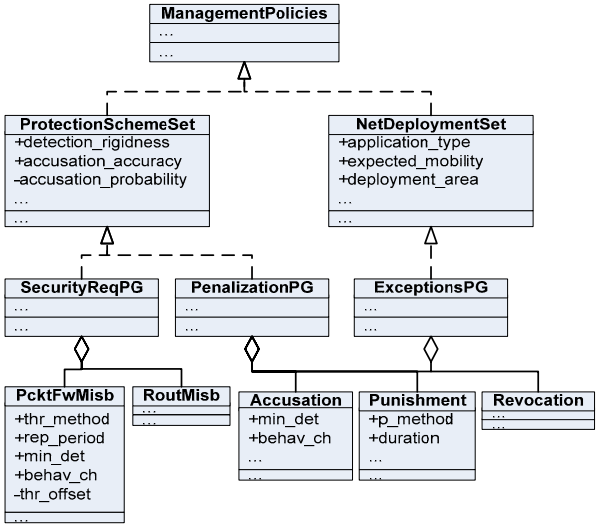

Figure 5. Policy types for adaptable misbehavior detection and accusation

The *protection scheme policy set* aims to facilitate the interaction of a managing entity with the underlying methods and algorithms that implement the misbehavior detection and

accusation. In a way, this set provides an abstraction to the manager, hiding the complex implementation details and offering two high-level parameters that can be dynamically manipulated: The *detection rigidness* expresses the level of tolerance the network should exhibit to misbehavior, while the *accusation accuracy* expresses the expected confidence in accusing and punishing nodes in the network. These parameters offer a mechanism to encapsulate the high-level management objectives as enforceable policies, by affecting the accusation probability P_A and determining the detection threshold offset.

The *security requirements group* can be further divided in packet forwarding misbehavior policies and routing misbehavior policies. The latter can be used to control an ad hoc routing protection scheme like SAODV [7]. The former policies are the ones that manipulate the parameters of our protection scheme and provide the desired versatility to change according to management objectives. It should be emphasized that the proposed policy set interaction not only provides the means to influence the protection scheme, but also provides the increased adaptability and parameterization offered from PBM systems by taking in mind network events and other related policy sets (e.g. penalization or network deployment groups).

The *penalization group* includes policies to control the node accusation, punishment, and revocation of punishment. Their aim is to refine the actions against those nodes detected as misbehaving by deciding when and how to penalize them. The importance of this group lays in the fact that it can offer a differentiation in the treatment of misbehavior. For example, assuming free vs. paid network access, paying customers can be offered a lower probability of being falsely accused. Accusation policies can provide additional conditions to expedite or delay node accusation while punishment policies control the type of actions against accused nodes. Revocation policies define if and when an accused node can be cleared of its current punishment, e.g. stop node isolation in 30 minutes.

Finally, the *network deployment* set can be used to express policies that prescribe the usage and purpose of the network and are not directly related to the proposed protection scheme. This set can encapsulate a priory management knowledge like the characteristics of the deployment area, the expected user mobility and the types of supported applications. Based on the above parameters, an *exceptions group* was created that includes similar policies with the penalization group. The purpose of this group is to differentiate the penalization of nodes depending on the deployment conditions that affect their normal behavior. For instance, areas with high mobility are expected to have a reduced packet delivery ratio, making mobile nodes more prone to misbehavior detection.

IV. Case Study and Evaluation

In order to evaluate the applicability and implications from the use of the proposed detection and accusation scheme, we first outline a case study scenario of a wireless ad hoc network. We then present the evaluation results regarding key aspects of our scheme and use the case study network to illustrate its effect regarding performance and management objectives.

A. Case Study Scenario

Wireless mesh networks are a prominent technology that has embraced the wireless ad hoc networking paradigm. A self-organizing network integrates a number of privileged nodes (mesh routers) that act as gateways for the rest of the participating wireless devices. In order to provide connectivity in areas out of the range of any gateway, such networks are reliant on the packet forwarding behavior of individual nodes along the created multihop paths. We assume the deployment of a mesh network as a case study scenario and consider a number of fixed nodes assigned the MN role, i.e. the devices under the direct control of a managing entity. A number of mesh routers is also deployed in the examined area and these nodes are assigned the CH role. For the purpose of this scenario, a number of user devices can be dynamically assigned the CH role, based on the adopted role-based framework and distributed algorithm (Section III). The rest of network nodes assume the CN role.

TABLE I. Illustrative Management Policies for Adaptive Protection

P	***Security Requirements Policy Group***
1	**on** {startup} **if** {-} **then** {{set_accus_accuracy(*high*)}, {set_detect_rigidness(*low*)}, {set_thresh_method(*simple*)}}
2	**on** {congestion} **if** {accused_count < *10%* } **then** {{ set_accus_accuracy (*med*)}, { set_detect_rigidness (*high*)}}
3	**on** {congestion} **if** {accused_ count > *60%* } **then** {{set_accus_accuracy (*high*)}, {set_detect_rigidness(*med*)}}
4	**on** {low_throughput} **if** { accused_ count < *10%* } **then** { set_thresh_method (*weighted*)}
	Penalization Policy Group
5	**on** {startup} **if**{-} **then** {set_accus_params(ch:=4,d:=2)}
6	**on** {startup} **if**{-} **then** {set_punishment(isolation)}
	Exceptions Policy Group
7	**on** {node_check} **if** {Node.role==CH}^{Node.owner==user} **then** { set_accus_params (ch:=*4*,d:=*3*}
8	**on** {node_check} **if** {Node.role==CH} **then** {set_ punishment (*warning*)}

All of the participating devices are expected to forward packets to facilitate the required multihop communication. In order to safeguard the operation of the mesh network, the managing entity integrates the proposed protection scheme and introduces appropriate policies to control and automate the detection and isolation of misbehaving users. As an proof of concept, some illustrative policies are included in Table I. The effect of our scheme on network performance is demonstrated in the simulations subsection that follows. For this case study, policies can be used to achieve the following goals:

1) Control the calculation method of the detection threshold

based on network conditions (e.g. congestion, P:2,3,4) or management objectives (e.g. detection accuracy, P1).

2) Control the offset based on the desired max probability of falsely accusing a well behaved node (P:1,2,3,4).
3) Differentiate the punishment enforcement based on the role, the application and the business model (P:5,6,7,8).

B. Simulation Results

We perform our simulations using the GloMoSim [24] simulation package. All values shown on our graphs, and their respective 95% confidence interval, are the result of averaging 20 simulation runs. The simulation parameters have been selected in order to obtain results that are meaningful when coupled with our case study scenario. Unless explicitly stated, our simulation parameters take the following values: i) nodes move according to the random waypoint mobility model with a speed randomly chosen with uniform distribution between 4m/sec and 6m/sec, yielding a mean node speed of 5m/sec (which is about the average speed of a car in a city center), and a standard deviation of 0.58m/sec, ii) the wireless transmission range of every node is 100 meters, iii) the node density is $5x10^{-4}$ nodes/m^2 iv) the link capacity is 2 Mbps, v) the MAC layer protocol is the IEEE 802.11 DCF, vi) the underlying routing protocol is AODV, vii) the total simulation time for each scenario is 1800 seconds, and viii) network isolation is the punishment enforced for any nodes accused of misbehavior. Roles are assigned to each node in the network following the adopted role-based framework (Section III). The execution of the distributed algorithm for the given fixed node density and variable node population yielded a ratio of approximately 1:3 for managing nodes (MN and CH) to node population. The number of MN is predefined as 3 and remains constant for all simulations, while the number of CHs depends on network size. Having in mind our case study, MN nodes can be operator-controlled mesh routers, while a number of user devices are selected as CHs to assist distributed management. In this section we first present a set of results showing how our protection scheme effectively improves the network performance in the presence of misbehaving nodes. Then the reaction time of our approach is shown for different network sizes and number of nodes. Finally, we demonstrate that the overhead introduced by our proposed scheme allows it to scale well.

An important parameter to evaluate the effectiveness of our approach is the network throughput gain that it can offer to networks affected by misbehaving nodes that drop packets in a probabilistic manner. In this section we present results that demonstrate that our protection scheme effectively improves the average network throughput as it detects and accuses misbehaving nodes in the network. For our first set of results, simulations for a 20 node network were run in an area of 240 000m^2 (489.9m x 489.9m). The network was set-up with 40% of their nodes misbehaving by dropping 80% packets. The misbehavior threshold $\alpha_{threshold}$, i.e. the maximum amount of packets that nodes are allowed to drop without being detected, is 60%. The sliding window size or number of behavior checks is ch = 4, and the minimum number of detections required for an accusation is d = 2. To improve graph readability we omit the calculated confidence intervals.

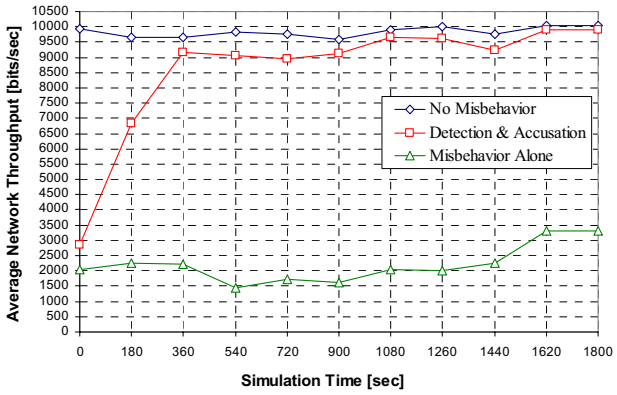

Figure 6. Average network throughput vs. simulation time (120 nodes)

Fig. 6 depicts the value of the average network throughput as the simulation time progresses from 0 to 1800 seconds. The graph displays a) networks without misbehaving nodes (No Misbehavior), b) networks with misbehaving nodes using our proposed protection scheme (Detection & Accusation), and c) networks with misbehaving nodes but with no means of defending themselves from any type of attack (Misbehavior Alone). As it can be seen, our protection scheme effectively mitigates the effects of packet forwarding misbehavior in a network and offers a substantial improvement on the average network throughput.

For Fig. 7 we consider a 60 node network over a terrain of 120 000m^2 (346.41m x 346.41m). Four groups of misbehaving nodes are set that drop 80%, 60%, 40% and 20% of packets respectively. Each of the groups consists of 10% of the total number of nodes in the network. The remaining network and algorithm parameters remain unchanged. For this set of results we assume a relaxed security policy with $\alpha_{threshold}$ fixed at 70% for the first half of the simulation. Half way through the simulation policies change the misbehavior threshold from a fixed value to be equal to the weighted average network behavior plus a pre-established offset (8%). It shows how our $\alpha_{threshold}$ = 70% policy is very ineffective as the network throughput does not improve significantly during the first half of the simulation. In the middle of the simulation the misbehavior threshold is changed by policies to be the weighted average network misbehavior plus an offset of 8%. This yields a value much lower than 70% which permits to detect and accuse previously undetected misbehaving nodes. As these nodes are isolated from the network the throughput becomes close to that of a network with well behaved nodes only.

Fig.8 shows the percentage of misbehaving nodes accused as the simulation time elapses. However, Fig. 8, unlike the

previous ones, allows us to see the reaction speed of our protection scheme to accuse misbehaving nodes. The curve corresponding to the network of Fig. 7 shows that initially (with $\alpha_{threshold}$ = 70%) only 25% of the misbehaving nodes are detected. However, once the misbehavior threshold is set to the weighted average network misbehavior plus the 8% offset the amount rapidly increases to reach 100%.

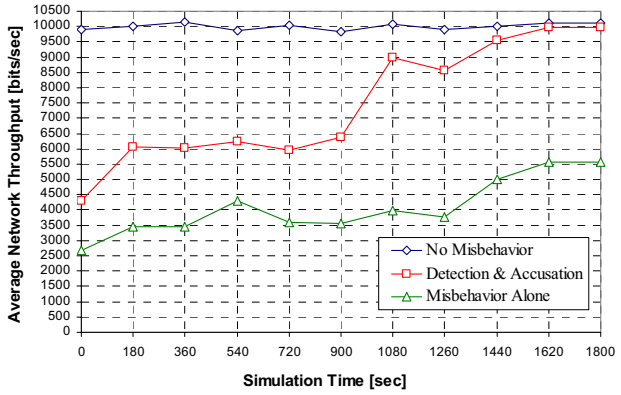

Figure 7. Average network throughput vs. simulation time (60 nodes)

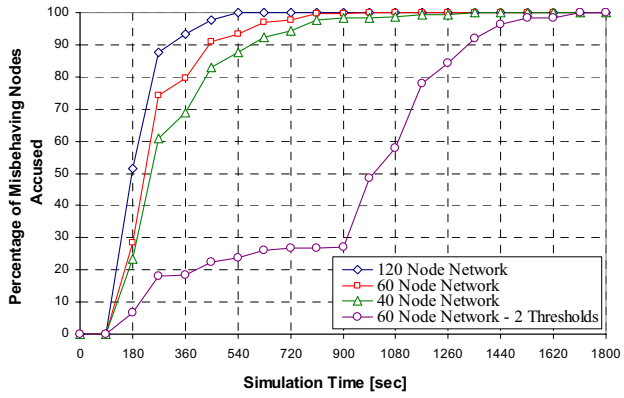

Figure 8. Percentage of misbehaving nodes accused vs. simulation time

Our final set of results presents the overhead produced by our protection scheme compared against the overhead produced by the underlying routing protocol (AODV) and the CBR traffic load (the application layer traffic). The information displayed corresponds to the 120 node network of Fig. 6. The mean node speed is increased from 0 to 20 m/sec in order to see the effect of mobility on our approach. Fig. 9 shows the total number of packets for three types of traffic: CBR traffic, AODV traffic and our proposed protection scheme traffic. The total packet number includes both generated and forwarded traffic. As it can be seen from the figure, our approach imposes a small overhead on the network which is also independent of node speed. This is expected since the traffic produced by our approach depends on how often behavior metrics are reported rather than on mobility issues. The reporting of information is controlled by the adopted policy-based framework (Section III) that has been investigated for scalability in [16]. Based on our results, we conclude that with the correct tuning of reporting parameters and its immunity to node speed, our protection scheme can scale well to medium and large wireless and mobile networks.

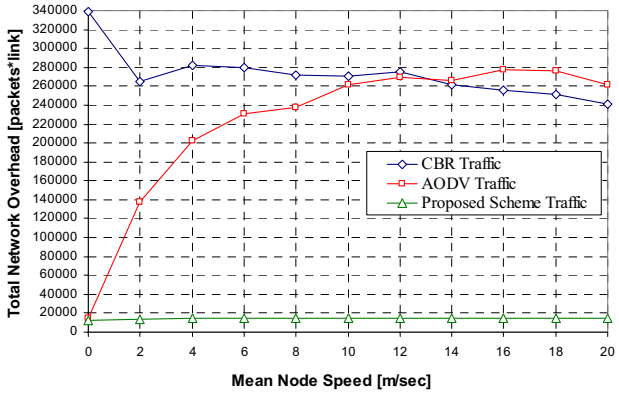

Figure 9. Total Network overhead vs. mean node speed (120 nodes)

V. Conclusions

In this paper we have presented an adaptable protection scheme that is capable of effectively detecting, accusing and punishing nodes that exhibit packet forwarding misbehavior in accordance with the changing network conditions and the management policies set by high-level entities. The effectiveness and efficiency of our approach was verified through an extensive set of simulations.

We have shown that by making use of policies at the management plane to control our scheme's detection and accusation parameters, the rigidness and accuracy of our approach can be customized in order to punish nodes that exhibit different levels of misbehavior in the network. Furthermore, different types of punishment can be established for nodes that execute different tasks or have different roles in the network. Finally, our proposed adaptable protection scheme eliminates the need for promiscuous listening (overhearing) since behavior evaluation is based on metrics directly collected by the actual communicating nodes.

This work focuses on providing protection to the data packet forwarding functionality only. However, within our approach there is scope to address the adaptable protection of routing protocols and part of our future work is to be aimed at this area. Also, in our scheme multiple policies and the network dynamic conditions can affect the values of the same managed object parameters. This introduces the risk of policy conflicts that we expect to address in our future work.

References

[1] S. Basagni, M.Conti, S.Giordano, I.Stojmenovic (Eds). Mobile Ad Hoc Networking. IEEE Press, 2004.

[2] M. Conti, S. Giordano, "Multihop Ad Hoc Networking: The Reality", IEEE Commun. Mag., Vol.45, No.4, pp.88-95, Apr.2007.

[3] H. Yang, J. Shu, X. Meng, S. Lu, "SCAN: Self-organized network-layer security in mobile ad hoc networks," IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, February 2006, pp. 261-273.

[4] I.F. Akyildiz, W. Xudong, "A survey on wireless mesh networks", IEEE Commun. Mag., Vol.43, No.9, pp. S23-S30, Sept. 2005.

[5] Y. C. Hu, D. B. Johnson, A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," Proc. 4th IEEE workshop on Mobile Computing Systems & Applications, pp. 3-13, June 2002.
[6] C. E. Perkins, P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Proc. ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, vol. 24, issue 4, pp. 234-244, 1994.
[7] M. Guerrero-Zapata, N. Asokan, "Securing ad hoc routing protocols," Proc. 3rd ACM Workshop on Wireless Security, pp. 1-10, 2002.
[8] C. E. Perkins, "Ad hoc on-demand distance vector (AODV) routing," Experimental, RFC 3561, July 2003.
[9] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," Proceedings of the 8th ACM International Conference on Mobile Computing and Networking, pp. 12-23, September 2002.
[10] S. Buchegger, J. Le Boudec, "Performance analysis of the CONFIDANT protocol," Proc. 3rd ACM Int. Symposium on Mobile Ad Hoc Networking & Computing, pp. 226-236, 2002.
[11] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, R. A. Olsson, "Detecting disruptive routers: a distributed network monitoring approach," Proc. 1998 Symposium on Security and Privacy, pp. 115-124, May 1998.
[12] R. Rao, G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," Proc. 2003 IEEE Global Telecommunications Conference, vol.5, pp. 2957-2961, 2003.
[13] S. Marti, T. J. Giuli, K. Lai, M. Baker, "Mitigating Routing Misbehavior in Mobile ad hoc networks," Proc. 6th ACM International Conference on Mobile Computing and Networking, pp. 255-265, August 2000.
[14] R. Boutaba, I.Aib, "Policy-based Management: A Historical Perspective", Journal of Network and Systems Management (JNSM), Vol.15, No. 4, Dec. 2007
[15] A.M.Hadjiantonis, A.Malatras, G. Pavlou, "A context-aware, policy-based framework for the management of MANETs", 7th IEEE Intl. Work. on Policies for Distributed Systems and Networks , pp.23-32 (Policy 2006)
[16] A. Malatras, A.M. Hadjiantonis, G. Pavlou, "Exploiting Context-awareness for the Autonomic Management of Mobile Ad Hoc Networks", Springer Journal of Network and System Management (JNSM), Vol. 15, No.1,pp.29-55, Mar.2007.
[17] R. Chadha et al, "Policy Based Mobile Ad hoc Network Management" 5th IEEE Intl. Work. on Policies for Distributed Systems and Networks.
[18] J. O. Kephart, D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, Vol. 36, No. 1, pp. 41-50, Jan. 2003.
[19] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", RFC2753, Informational, Jan.2000.
[20] J. Wu, H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks", in Proc. 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Washington, USA, pp.7-14, 1999.
[21] O. F. Gonzalez, M. Howarth, G. Pavlou, "Detection and accusation of packet forwarding misbehavior in mobile ad-hoc networks," Journal of Internet Engineering, vol. 2, no. 1, July 2008, pp. 181 – 192.
[22] T.Clausen, P.Jacquet (eds), "Optimized Link State Routing Protocol (OLSR)", RFC3626, Experimental, Oct.2003
[23] B. Moore, "Policy Core Information Model (PCIM) Extensions", RFC3460, Standards Track, Jan.2003.
[24] M. Takai, L. Bajaj, R, Ahuja, R. Bagrodia M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," Technical report 990027, UCLA, Computer Science Department, 1999.