A Framework for Detecting Selfish Misbehavior in Wireless Mesh Community Networks

Fabio Martignon
Department of Information Technology and Mathematical Methods
University of Bergamo
fabio.martignon@unibg.it

Stefano Paris
Department of Electronics and Information
Politecnico di Milano
paris@elet.polimi.it

Antonio Capone
Department of Electronics and Information
Politecnico di Milano
capone@elet.polimi.it

ABSTRACT

Wireless Mesh Networks (WMNs) have recently emerged as a flexible and low-cost extension of wired infrastructure networks. They consist of mesh routers and clients, where mesh routers are almost static and form the backbone of the WMN.

The complete absence of an infrastructure and the flexibility provided by the wireless mesh technology has fostered the development of new network paradigms like Wireless Mesh Community Networks. Such networks are usually composed of heterogeneous mesh routers managed by different users (a subset of participants to the community), that collaborate to extend the network coverage. However, in such environment some participants can exhibit selfish behaviors, by dropping selectively the packets sent by other mesh routers, in order to prioritize their own traffic and increase their network utilization.

In this paper we propose a complete scheme to detect selfish behavior of the mesh routers that participate to the community network. Each node evaluates the trustworthiness of the other mesh routers by combining the direct observations on the relaying behavior of neighbor nodes with the trust information provided by other mesh routers. The proposed scheme has been integrated in the AODV routing protocol, and tested in several network scenarios.

The numerical results show that our scheme provides a high detection accuracy, even when a high percentage of network nodes provide false trust values (bad-mouthing attack).

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms

Design, Security

Keywords

Wireless Mesh Community Networks, Trust, Reputation

1. INTRODUCTION

Wireless Mesh Networks (WMNs) have emerged as a technology for next generation wireless networking. The complete absence of a fixed infrastructure has promoted new network paradigms like Wireless Mesh Community Networks (WMCNs) [1], which provide a viable alternative to municipal wireless networks for consumers, but has also introduced new problems that are hard to overcome with current communication protocols.

Existing security algorithms and protocols developed for WMNs assume that each wireless mesh router participates honestly in the execution of networking procedures. This assumption is valid only in a network managed by a single trusted authority. However, a Wireless Mesh Community Network can be formed by a group of independent mesh routers owned by different service providers or individuals. A selfish user that provides connectivity through his own mesh routers to other network nodes might try to greedily consume the available bandwidth by favoring his own traffic while selectively dropping others' [2].

Trust and reputation frameworks can be implemented at the network layer in order to detect selfish behaviors or stimulate the cooperation among different routers, since they enable the collaborative evaluation of the behavior of all mesh routers, thus permitting to improve the overall availability of the Wireless Mesh Community Network. This evaluation could take into account both *direct* observations on the behavior of neighbor nodes and the *indirect* information provided by other mesh routers. This latter must also be discounted by a trust degree that the node owner has in other providers in order to filter false trust scores.

Even if trust and reputation are closely related terms, there is a distinction between the two concepts: *reputation* is a perception that a node creates through past interactions with other nodes, whereas *trust* is a subjective expectation that a node has about the future actions that will be performed by other nodes [3]. The trust of other nodes can be therefore evaluated as a function of their reputation and other factors, such as the time elapsed since the reputation was last measured.

The trust related to a specific node is updated periodically with both direct observations of the action outcome and the indirect trust provided by other nodes (i.e. the trust other nodes have in that specific node).

In this paper we propose a complete scheme to detect selfish behavior of the mesh routers that participate to the community network, based on a novel trust and reputation management system. Each node evaluates the trustworthiness of the other mesh routers by combining the direct observations on the relaying behavior of neighbor nodes with the trust information provided by other mesh routers.

The proposed framework is composed of three elements: a *watchdog* mechanism able to distinguish between selfish and cooperative actions, a *protocol* to exchange trust ratings among the network nodes, and a *trust model* for quantifying the nodes trustworthiness.

We implement the watchdog mechanism by setting each node in promiscuous mode, so that the mesh router can evaluate the relaying behavior of its neighbors by analyzing the eavesdropped traffic, that is, by verifying that neighbors are actually forwarding packets, and not dropping them [4].

Since the message exchange required by the protocol can consume considerable network resources, we integrate our detection scheme in the AODV routing protocol by adding new routing messages and procedures to gather trust ratings by other network nodes and compute the global trust of the other mesh routers. However, we observe that the proposed framework can be implemented in any routing protocol and used to choose the most trustworthy path, when more alternatives are available.

Finally we proposes a novel trust model, based on the well-known vector model, that enables the aggregation of all the trust scores related to a specific mesh client in a unique trust value.

We evaluate numerically the proposed framework by simulating typical network scenarios. The results show that our scheme provides a high detection accuracy, even when a high percentage of network nodes provide false trust values (bad-mouthing attack).

The paper is structured as follows: Section 2 discusses related work. Section 3 presents the network model and assumptions, as well as the adversary model considered in our work. Section 4 illustrates the proposed detection scheme. Section 5 provides a numerical evaluation of the proposed framework. Finally, conclusions and directions for future work are presented in Section 6.

2. RELATED WORK

Reputation and trust-based systems have been applied to several distributed systems as a valuable method to enforce the collaboration among network nodes. A reputation-based scheme for detecting anomalous behaviors of network nodes is proposed in [5]. The proposed algorithm analyzes the temporal and spatial properties of direct and indirect observations to detect anomalous misbehavior and decrease the false positive rate. The work [6] presents a distributed protocol to establish the trustworthiness of network nodes and spread such information through a secure acknowledgment scheme.

Different theoretical approaches have been exploited to model the trust related to the forwarding behavior of network nodes. In [7] the authors propose an information theoretic framework to quantitatively measure trust and model trust propagation in ad hoc networks. The works [8] and [9] adopt a probabilistic approach to model the reputation and the trustworthiness of network entities. These two latter have led the design of the reputation-based framework proposed in [10]. The authors of [11] exploit fuzzy logic to deal with the uncertainty related to the evaluation of packet forwarding and recommendations trustworthiness.

In credit-based approaches a network node is rewarded when it forwards the packets sent by other nodes. In [12] the authors propose a distributed algorithm based on the concept of reciprocity among nodes, where the credit is represented by the amount of traffic directly or indirectly forwarded by other network nodes. SPRITE [13] defines a rewarding mechanism which enforces the forwarding as the best strategy. The proposed solution is based on a centralized trusted third party that charges or rewards the forwarding nodes on the basis of the collected receipts. In [14] the authors propose two forwarding approaches, the Packet Purse Model (PPM) and the Packet Trade Model (PTM), through which the intermediate nodes are rewarded. In the former protocol the packet carries the necessary credits paid by the source to each intermediate node, whereas in the latter the packet is traded for credits by intermediate nodes. Both protocols assume the existence of a tamper resistant module which is liable for all the cryptographic and payment procedures. This assumption cannot be extended to wireless mesh community networks, since each community participant directly manages his mesh routers.

Besides the problems discussed above, all credit-based schemes are based on the assumption that each node moves inside the network and thus has the opportunity to be selected as relaying node. However, in a WMCN the mesh routers are almost static, and leaf nodes might not earn enough credits by forwarding only the traffic of the external customers.

Therefore, the applicability of all the above schemes is very limited in a wireless mesh community scenario.

3. SYSTEM MODELS AND ASSUMPTIONS

In order to specify the WMCN scenario we are dealing with, we present the communication and threat models considered in our architecture, as well as the definitions and assumptions we adopt in the design of our detection techniques.

3.1 Assumptions

We adopt the following definitions and assumptions:

- all network devices communicate with each other using the wireless medium, in particular the IEEE 802.11 MAC protocol.

- All mesh routers are endowed with an omnidirectional antenna for backbone communications. All backbone links use the same wireless channel.

- All wireless links established between any two nodes are symmetric, and we do not consider in this paper the issues deriving from asymmetric channels.

3.2 Network Model

This work considers a wireless mesh community network composed of two different types of devices:

- the mesh routers that form the infrastructure of the wireless mesh community network. These nodes are managed and maintained by different community users.

- The customer devices that are only interested in the services provided by the WMCN (e.g., Internet access).

The mesh router owners can connect to the backbone network with their wireless devices, whereas the customers can only access the WMCN services through the mesh routers. In fact, the mesh routers can be endowed with another wireless interface and operate like access points of a WLAN in order to provide access to generic customers that do not participate to the community.

Community users and customers may be charged different fees to access the WMCN services. As a consequence, these services must satisfy Quality of Service requirements, and penalties can be envisaged if QoS requirements are violated.

Since the WMN architecture we consider has a hierarchic structure (wireless mesh routers are in fact dedicated nodes which are deployed to offer backhaul services), we suppose the existence of a subset of community participants that are liable for all management tasks.

3.3 Adversary Model

We assume that the users of the wireless mesh community network are selfish, but they do not act as malicious users. To cope with the latter case, several architectures based on standard cryptographic primitives can be used [15, 16].

Mesh routers owned by community users perform all the procedures required by network protocols, but some of them behave selfishly towards the nodes they serve. In fact, all the community users have two opposed interests: on one hand they compete against the customer devices which they serve for the available network bandwidth provided by the mesh routers, since they have to share the capacity of their outgoing wireless link established with other mesh routers of the WMCN.

On the other hand, mesh routers have an incentive to serve a large number of users in a fair way, since we assume they are rewarded by the mesh community network considering both the number of served customers and their satisfaction. The rewarding policy applied by the mesh community network is out of the scope of this paper.

The community users that manage mesh routers can set firewall rules on their devices to drop almost all packets sent by other participants or customer stations, or limit the maximum transmission rates available to the served devices. Note that the dropping attack can be considered as a special case of the latter attack, in which the rate of the served device is close to zero. The packet dropping attack can be performed by a mesh router both on the outgoing and ingoing traffic. Therefore, the other network devices that participate to the wireless mesh community network must evaluate the behavior of an intermediate mesh router, controlling if the neighbor wireless mesh routers relay all the network traffic after having confirmed its reception.

As suggested by reputation and trust framework applied to wireless ad hoc networks [4], the watchdog installed on every device performs such activity by observing the channel to evaluate if the intermediate node forwards the received packets.

In order to subvert the detection system, selfish community users can provide dishonest recommendations to other network devices. Such attack is also known as *bad-mouthing attack* [17]. Since indirect trust plays an important role in the evaluation of the global trust, the trust model has to be designed in such a way that mesh routers assign low weights to the recommendations provided by network nodes with low recommendations' trustworthiness. As a result, if a mesh router has low recommendations' trustworthiness, its recommendations will have minor influence on the computation of the global trust in other network devices.

4. A FRAMEWORK FOR DETECTING SELFISH MISBEHAVIOR

In this Section we describe the framework we propose to evaluate the behavior of devices that belong to the wireless mesh community network. We first describe the architecture of our detection framework, then illustrate the trust model and the algorithm used by each node to compute the trust in other mesh routers.

4.1 Architecture

The architecture of the detection system is composed of two elements: a watchdog that detects selfish behavior (packet dropping), and a trust model used by each mesh router to compute the global trust in other network nodes.

Figure 1 sketches the architecture of the proposed detection system implemented on all mesh routers of the WMCN.

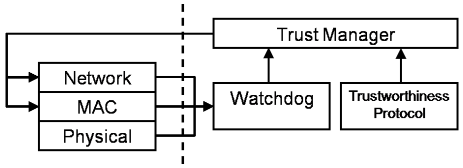


Figure 1: Architecture of the proposed detection system.

The *watchdog* gathers information from different protocol layers to distinguish between cooperative and selfish actions of neighbor nodes. In our architecture, the watchdog maintains for each neighbor the number of successfully received packets, that is, the number of frames to which it has replied with an acknowledgement, p_a, and the number of forwarded packets with the same source address of the acknowledged packets, p_f. The ratio between these two values, $\frac{p_f}{p_a}$, represents the direct trust the node has in each neighbor as relaying node, and it is used by the *Trust Manager* to compute the global trust.

To illustrate how the direct trust is evaluated by the watchdog, let us refer to the example network scenario shown in Figure 2, where solid and dotted lines represent the transmission of packets and acknowledgments, respectively. When mesh router $N1$ receives from $N2$ the acknowledgment for a previously sent packet, $N1$ monitors the wireless channel until it hears the retransmission of the same packet performed by $N2$ (towards $N3$, see Figure 2(a)). If such retransmission does not occur before a timeout expires, $N1$ will conclude that $N2$ has not forwarded its packet and increment only the counter of the number of acknowledged packets, p_a; otherwise it will increment also the number of forwarded packets, p_f. The *timeout* parameter is tuned to take into account processing and transmission delays.

In our architecture a mesh router evaluates the direct trust in its neighbors considering all the packets transmitted to them, i.e. all the packets sent by the nodes inside its transmission area that it can correctly decode. As shown in Figure 2(b), $N1$ considers also the packets transmitted by $N4$. If $N1$ does not hear the retransmission of the acknowledged packet sent by $N4$ before the timeout expires, it will conclude that $N2$ has dropped it and it will update only the number of packets transmitted to $N2$.

Note that the watchdog has all the necessary elements (i.e., the MAC and network headers) to perform the analysis described above. In fact, when a network card is set in monitoring mode, all the decoded frames transmitted on the wireless channel can be read and analyzed by the watchdog process.

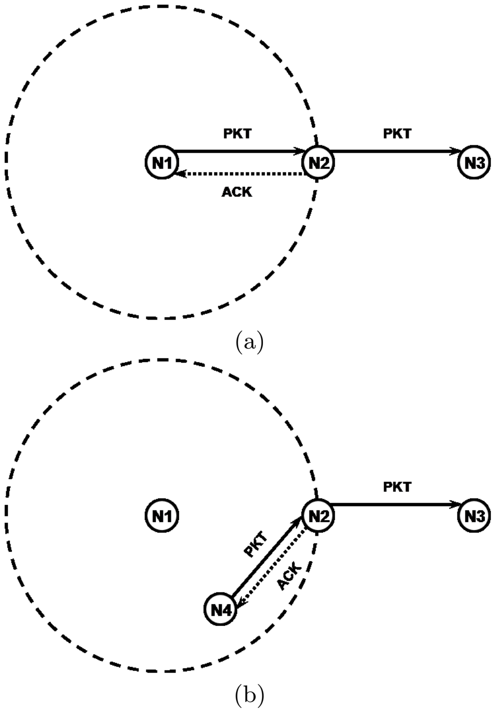


Figure 2: Example of detection performed by the watchdog installed on node $N1$

The *trustworthiness protocol* is used by each node to exchange with other mesh routers the direct trust in its neighbors. As explained in Section 4.2, such values represent the indirect trust the receiving node has in the nodes which the values refer to, and are used to improve the knowledge that the receiving mesh router has about the trustworthiness of other network nodes.

At the end of a test interval, the *trust manager* computes the global trust in every network node as a function of the direct and indirect trusts provided by the *watchdog* and the *trustworthiness protocol*, respectively. The global trust can be used to exclude the mesh routers that have a low global trust or to modify the standard behavior of MAC and network protocols, like routing. For example, the global trust can be used to define a new routing metric in order to select the most trustworthy among multiple paths with the same hop count or create a cluster whose route paths will include only safe nodes [18].

4.2 Trust Model

In computer networks there exist two methods to establish the global trust value assigned to a network entity. First, an entity i can assess a *direct trust* rate in another network entity j through direct observations of its behavior. Such value can be integrated with the *indirect trust* value that is computed using the recommendations on entity j provided by other network entities $k \neq i, j$.

The basic notation used in the rest of the paper is reported in Table 1.

Table 1: Basic notation used in the paper

Parameter	**Definition**
μ_{ij}	Direct trust node i has in node j
ν_{ikj}	Indirect trust on node j provided by node k to node i
σ_{ik}	Recommendations' trustworthiness node i has about node k
τ_{ij}	Global trust node i has in node j
ρ_{ij}	Reputation of node j computed by node i
N	Set of mesh routers, $\|N\| = n$

In our framework, each mesh router i represents the direct trust of all other mesh routers that participate to the WMCN as an $(n-1)$ vector $\overrightarrow{\mu_i} = (\mu_{i1}, \mu_{i2}, ..., \mu_{i(n-1)})$, where each element μ_{ij} represents the percentage of packets that node i has heard to be forwarded by node j. Note that some components of the vector can be null if mesh router j is outside the reception area of node i, while the element $\mu_{ii} = 1$ since each mesh router trusts itself, by definition.

The *indirect trust* that mesh router i has in a neighbor mesh router j is obtained as a function of the direct trust that every mesh router k has in j. In particular, we represent the indirect trust provided by node k to node i as an $(n-2)$ vector $\overrightarrow{\nu_{ik}} = (\nu_{ik1}, \nu_{ik2}, ..., \nu_{ik(n-2)})$, whose component ν_{ikj} is the direct trust that mesh router k holds in mesh router j, i.e. $\nu_{ikj} = \mu_{kj}$.

Thus, mesh router i can compute recommendations' trustworthiness of node k evaluating the similarity between the vector $\overrightarrow{\tilde{\mu_i}}$ (i.e. the vector $\overrightarrow{\mu_i}$ without the component i and k) and that provided by k, $\overrightarrow{\nu_{ik}}$, according to expression (1):

$$\sigma_{ik} = \frac{\overrightarrow{\tilde{\mu_i}} \cdot \overrightarrow{\nu_{ik}}}{\parallel \overrightarrow{\tilde{\mu_i}} \parallel \cdot \parallel \overrightarrow{\nu_{ik}} \parallel} = \frac{\sum_{j=1}^{n-2} \tilde{\mu_{ij}} \nu_{ikj}}{\sqrt{\sum_{j=1}^{n-2} \tilde{\mu_j}^2} \sqrt{\sum_{j=1}^{n-2} \nu_{ikj}^2}} \qquad (1)$$

The recommendations' trustworthiness of a mesh router k represents, therefore, the cosine of the angle between the trustworthiness's vectors of mesh routers i and k. Such value provides an indication of the distance between the trust scores of the two mesh routers, and as a consequence such technique permits to detect the bad-mouthing attack. In fact, if node k provides a trust vector orthogonal to that computed by i ($\overrightarrow{\tilde{\mu_i}} \perp \overrightarrow{\nu_{ik}}$), then i will have no trust in the trustworthiness scores provided by k.

The similarity permits to filter out all recommendations that are likely provided by bad-mouthing nodes, whose interest is to increase the reputation of their friendly mesh routers to the detriment of cooperative ones.

The similarity between the trust vectors of nodes i and k, σ_{ik}, is computed considering only the elements which represent the mesh routers that have interacted both with node i and k (i.e., the mesh routers that have been inside the reception area both of i and k, and that have been controlled by these latter).

If mesh router k has no experience on node j as relaying node (or mesh router i has no experience on node j), since mesh router j has never been inside the reception area of node k, we set $\nu_{kj} = \mu_{ij}$. In this way, the component kj is not considered when computing the discount factor of the observation provided by mesh router k (i.e., the trustworthiness that mesh router i holds in recommendations provided by k on node j). In fact, in the previous computation the elements that are equal in the two vectors do not contribute to decrease the mesh router recommendations' trustworthiness σ_{ik}.

Node i computes the trust of mesh router j considering both direct and indirect trust (Equation (2)): the former is a function of direct observations, whereas the latter has to consider the scores of all the nodes that have provided their direct trust in mesh router j to node i in the last interval. We use a convex combination of direct and indirect trust, where α is a weight used to provide more importance to the former or the latter; t represents the current iteration (or interval).

$$\begin{aligned} \tau_{ij}(t) &= \alpha\mu_{ij} + (1-\alpha)\left[\frac{1}{\Sigma}\sum_{k=1}^{n-2}\sigma_{ik}\nu_{kj}\right] \\ \Sigma &= \sum_{k=1}^{n-2}\sigma_{ik} \end{aligned} \tag{2}$$

At the end of each interval, mesh router i updates the reputation of node j according to Equation (3), where β is a weight that balances fresh and old observations.

$$\begin{cases} \rho_{ij}(t) = \tau_{ij}(t) & \text{if } t = 1 \\ \rho_{ij}(t) = \beta\rho_{ij}(t-1) + (1-\beta)\tau_{ij}(t) & \text{if } t > 1 \end{cases} \tag{3}$$

Note that the direct trust, $\overrightarrow{\mu_i}$, and the recommendations' trustworthiness, σ_{ik}, are evaluated in each interval. However, the computation of the recommendations' trustworthiness (i.e. the similarity) can take into account the new information on the reputation of the mesh routers computed at the end of the previous interval, i.e. $\rho_{ij}(t)$. Therefore, the recommendations' trustworthiness can be evaluated by substituting a linear combination of $\rho_i(t)$ and $\overrightarrow{\tilde{\mu_i}}(t-1)$ for $\overrightarrow{\tilde{\mu_i}}$ in Equation (1), according to the following expression, where γ is a weight that balances direct trust and reputation:

$$\tilde{\mu_{ij}}(t) = \gamma\tilde{\mu_{ij}}(t) + (1-\gamma)\rho_{ij}(t-1) \tag{4}$$

4.3 Trust Computation Algorithm

The trust computation algorithm implements the message exchange and all the procedures necessary to compute the trust and reputation of all other network nodes, as described by the Trust Model. A detailed description of the proposed algorithm, performed by the trust manager of a generic node i, is listed in Algorithm 1. The algorithm receives as input the parameters α, β, γ and $Timeout$, which defines the duration of a trust computation interval. In the simulations we have set $\alpha = 0.8, \beta = 0.8, \gamma = 0.5$ and $Timeout = 10s$, which provided good performance. We plan to gauge the sensitiveness of the proposed detection framework to the parameters' setting in future works.

At the end of a pre-selected interval (parameter Timeout), in which the watchdog installed on the generic mesh router i gathers information about the forwarding behavior of its neighbors, the trust manager broadcasts a trust request to know the direct trust vectors of the other nodes that could have recently interacted with node i's neighbors. To this end, the Time To Live (TTL) of the request message is set to 2, since the network nodes which could have observed the behavior of i's neighbors are at most two hops away from i.

Algorithm 1 Trust Computation Algorithm

```
Require: $Timeout, (\alpha, \beta, \gamma) \in [0,1]^3$
  $t = 1$
  $\overrightarrow{\mu_i} = \overrightarrow{-1}$
  for all  $k \in N \setminus \{i\}$  do
      $\overrightarrow{\nu_{ik}} = \overrightarrow{-1}$
  end for
  loop
      Sleep(Timeout)
      for all  $j \in N(i)$  do
          Watchdog.stats($p_f$, $p_a$, $j$)
          $\mu_{ij} = \frac{p_f}{p_a}$
          if  $t == 1$  then
              $\tilde{\mu_{ij}} = \mu_{ij}$
          else
              $\tilde{\mu_{ij}} = \gamma\tilde{\mu_{ij}} + (1-\gamma)\tau_{ij}$
          end if
      end for
      Broadcasts a trust request with TTL = 2
      for all  $r \in \{TREP\}$  do
          $k$ = SourceOf($r$)
          $\sigma_{ik} = \frac{\overrightarrow{\tilde{\mu}_i} \cdot \overrightarrow{\nu_{ik}}}{\|\overrightarrow{\tilde{\mu}_i}\| \cdot \|\overrightarrow{\nu_{ik}}\|}$
      end for
      for all  $j \in N \setminus \{i\}$  do
          if  $\mu_{ij} \neq -1$ then
              $\tau_{ij} = \alpha\mu_{ij} + (1-\alpha)\left[\frac{1}{\Sigma}\sum_{k=1}^{n-2}\sigma_{ik}\nu_{kj}\right]$
          else
              $\tau_{ij} = \left[\frac{1}{\Sigma}\sum_{k=1}^{n-2}\sigma_{ik}\nu_{kj}\right]$
          end if
          if  $t == 1$  then
              $\rho_{ij} = \tau_{ij}$
          else
              $\rho_{ij} = \beta\rho_{ij} + (1-\beta)\tau_{ij}$
          end if
      end for
      $t++$
  end loop
```

Each node that receives a trust request sends back to the source a trust reply, which contains the direct trust scores the replier has evaluated on other network nodes. The trust reply carries at most $2(n-2)$ elements, since a generic node k provides neither the direct trust in the node that sent the request nor the direct trust in itself.

Once node i has waited for enough time, it computes the recommendations' trustworthiness of each sender, evaluating the similarity between its own direct trust vector and that provided in the reply. The recommendations' trustworthiness is computed for all the received trust replies, stored in the $\{TREP\}$ set.

Finally, the trust manager can compute the trust of every network node as a function of direct and indirect trust ($\overrightarrow{\tau_i}$), and update the reputation considering the entire history of the nodes ($\overrightarrow{\rho_i}$).

For the sake of brevity we do not show the operations performed by the trust manager when only one node between i and k has directly observed the behavior of node j. As described in Section 4.2, in this case the trust score is not considered in the evaluation of σ_{ik}. However, the indirect trust in node j provided by all other mesh routers is used in the computation of the global trust and reputation, even if node i has no direct trust in j.

The proposed algorithm has been integrated in the AODV routing protocol by adding new routing messages (a trust request and a trust reply) as well as novel procedures to gather trust ratings by other network nodes and compute the global trust in other mesh routers. We name AODV-T (AODV with Trust) the modified version of AODV.

5. NUMERICAL RESULTS

In this Section we present and discuss the numerical results obtained testing the proposed detection framework with Network Simulator. We first describe the simulation settings, then we measure the performance of our detection scheme.

5.1 Network Configuration

In our simulations, we consider a typical WMCN composed of 40 mesh routers placed over an area of $2000m \times 2000m$ to form a 5×8 grid topology, illustrated in Figure 3. The maximum channel capacity is set to 54 Mbit/s. All nodes employ the IEEE 802.11g standard MAC and use the same wireless channel, since ns v.2 does not support natively multi-channel or multi-interface wireless nodes. As routing agent we use AODV-T, the modified version of AODV that implements the trustworthiness protocol and the trust management functions illustrated in Section 4. Table 2 summarizes the parameters used in our simulations.

Table 2: Simulation Parameters

Parameter	Value
MAC	IEEE 802.11g
Routing Protocol	AODV-T (AODV with Trust)
Transmission Range	250 m
Receiving Range	550 m
CBR Rate	400 kbit/s
Packet length	1000 byte

We analyze the performance of the proposed detection scheme, varying both the number of *selfish* and *bad-mouthing* nodes. In particular, we consider three different network scenarios including: (1) only selfish nodes, (2) selfish and bad-mouthing nodes and (3) selfish with colluding bad-mouthing nodes. The first scenario provides an upper bound for the performance of our detection scheme, since it assumes that selfish mesh routers provide legitimate trust values of other network nodes. The second scenario considers a more realistic type of attack, since selfish nodes, besides dropping packets sent by other nodes, provide false trust values to other requesting mesh routers. Finally, the latter scenario takes into account also collusion between different nodes: selfish nodes perform only the packet dropping attack but they provide legitimate trust values, while a second group of nodes lie about trust values (i.e., such nodes performs the bad-mouthing attack). For the sake of brevity, in this scenario we illustrate only numerical results where the fraction of bad-mouthing nodes is fixed and equal to 30%. This attack is more serious than the second, since the trustworthiness of the source node that provides the trust reply is not considered in the indirect trust computation, and therefore selfish nodes cannot change the opinion that other nodes have on them providing false trust information about themselves.

As shown in Figure 3, we divide the grid network into three subareas. The central area contains the selfish mesh routers that drop the packets sent by other nodes with a drop rate ranging between 10% and 90%. Each source node generates a CBR traffic with a rate equal to 400 kbit/s towards the corresponding destination node at the right end of the same row (e.g., node 1 transmits to node 5). The number of CBR connections is therefore 8. The packet dimension is equal to 1000 bytes.

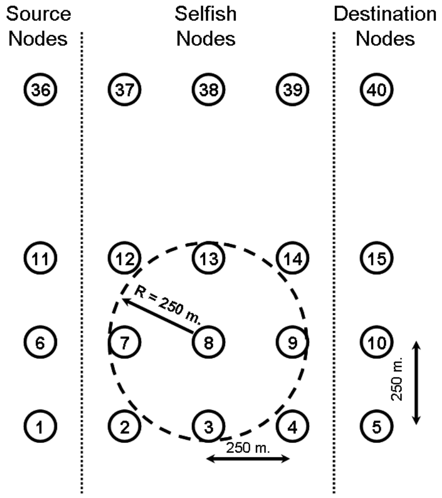


Figure 3: Grid Network.

5.2 Analysis of the Results

We first measure the average throughput achieved by the CBR connections. Figure 4 illustrates such performance figure as a function of the fraction of selfish nodes as well as of the packet drop rate. The presence of a relatively low number of selfish nodes (less than 15%) can lead to severe throughput degradation, thus making the leaf nodes (who provide the access service) responsible for the violation of QoS requirements. Such observation has driven us to design the proposed detection framework.

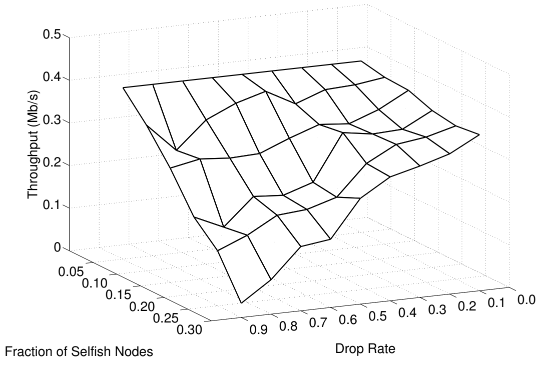
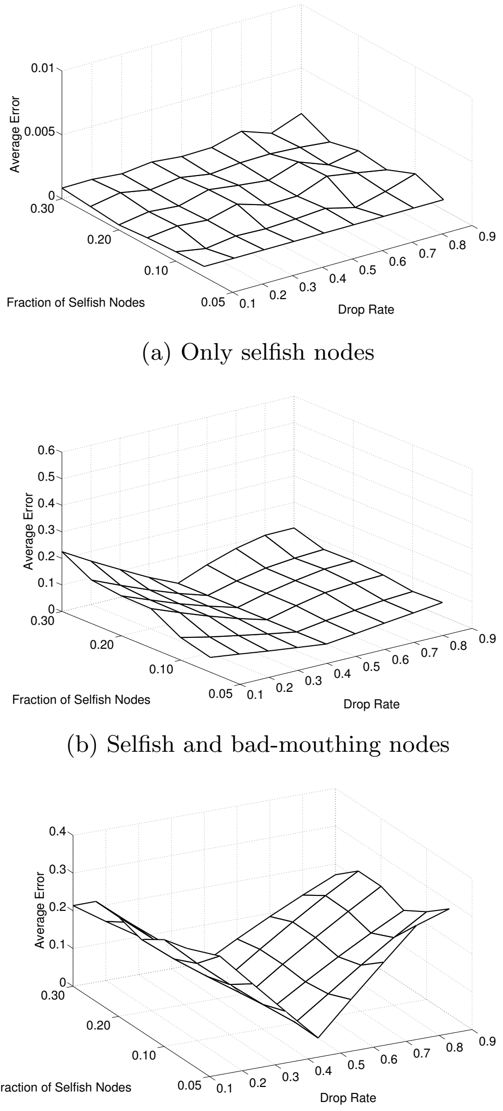


Figure 4: Average throughput of the CBR connections in the network of Figure 3, as a function of the fraction of selfish nodes and the drop rate.

In order to gauge the robustness of the proposed framework, we measured the average absolute difference between the real reputation of the selfish mesh routers and that computed by the honest nodes that use the proposed trust model (average error). The numerical results for the three scenarios described previously are illustrated in Figure 5.

We observe that when only selfish nodes are present, as in the first scenario we consider (Figure 5(a)), our proposed scheme achieves a very high accuracy for every fraction of selfish nodes and for all drop rates, since no mesh router provides dishonest trust values to other network nodes.

On the other hand, the V-shaped plot in Figures 5(b) and 5(c) is due to the effect of the bad-mouthing action. In fact, a bad-mouthing node provides a trust vector equal to $1 - \overrightarrow{\mu_j}$ to requesting nodes. Therefore, when the drop rate is equal to 0.5 the average error is null, since both honest and bad-mouthing mesh routers provide the same values.



(a) Only selfish nodes

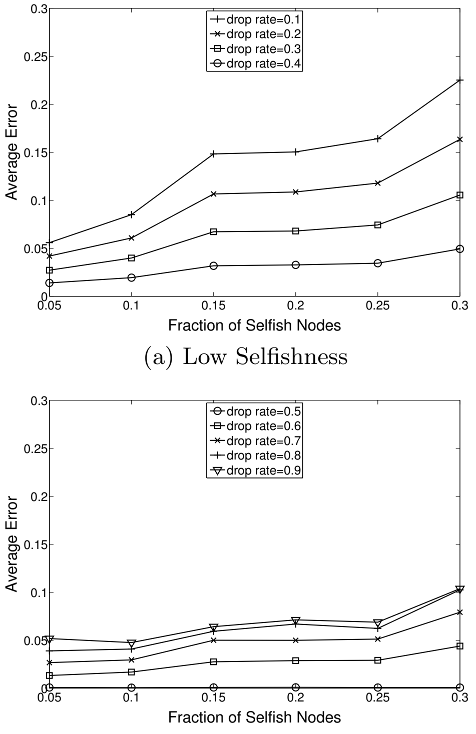
(b) Selfish and bad-mouthing nodes

(c) Selfish with colluding bad-mouthing nodes (fraction of bad-mouthing nodes = 0.3)

Figure 5: Average difference between the true reputation of selfish nodes and that computed by other mesh routers.

Figures 6(a) and 6(b) illustrate in detail the curves of the second network scenario (selfish and bad-mouthing nodes) for different drop rate values. For the sake of clarity, Figure 6(a) shows the curves obtained for low drop rates (0.1-0.4), while Figure 6(b) reports those obtained for high drop rates (0.5-0.9). It can be observed that the average error increases when the number of dishonest nodes increases (recall that, in this scenario, selfish nodes are also bad-mouthing), and that the higher is the node selfishness, the lower is the detection error of our framework.

We observe that our proposed framework is sufficiently accurate when the drop rate increases, thus representing an effective solution to improve the performance of a WMCN where users exhibit selfish behaviors.



(a) Low Selfishness

(b) High Selfishness

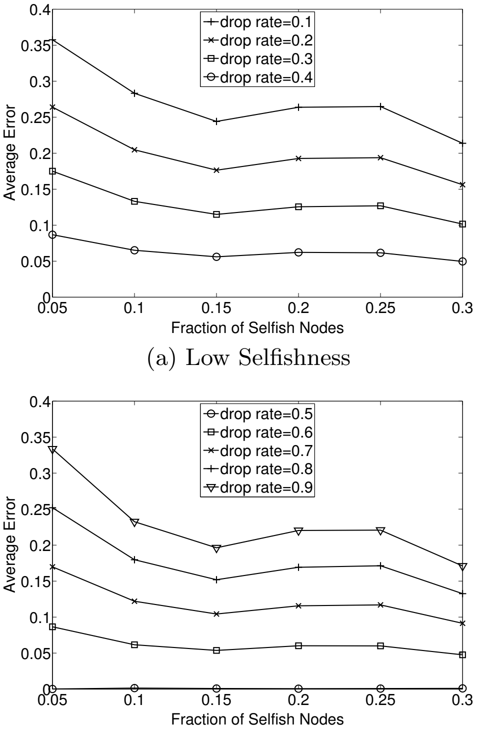
Figure 6: Average difference between the true reputation of selfish nodes and that computed by other mesh routers for the selfish and bad-mouthing scenario.

Figures 7(a) and 7(b) illustrate the same curves for the third scenario, where selfish with colluding bad-mouthing nodes coexist.

Such collusion increases the average detection error, since honest nodes use all the values provided by bad-mouthing nodes when they compute the indirect trust in selfish nodes. In particular, when the number of bad-mouthing nodes is higher than the number of selfish nodes, as shown in Figure 7, the average error decreases when the number of dishonest nodes increases. Therefore, bad-mouthing nodes masquerade the behavior of selfish nodes when these latter are a minority group. However, as the number of selfish nodes increases, the detection accuracy improves (the average error decreases) since a larger number of honest nodes can observe the misbehavior and inform the other network nodes.

6. CONCLUSION

In this paper we proposed a complete scheme to detect selfish behaviors of mesh routers that participate to a Wire-



(a) Low Selfishness

(b) High Selfishness

Figure 7: Average difference between the true reputation of selfish nodes and that computed by other mesh routers for the "selfish with colluding bad-mouthing scenario". The fraction of bad-mouthing nodes has been fixed to 0.3.

less Mesh Community Network. Our system provides an effective solution to detect network nodes that drop packets instead of forwarding them.

We implemented the proposed framework in Network Simulator, integrating it in the AODV routing protocol, and we tested it in typical network scenarios.

Numerical results show that our scheme offers a very high detection accuracy, even when a high percentage of network nodes provide false trust values.

Future research issues include the design of a routing metric to choose the most trustworthy path when more alternatives are available, and the study of a more sophisticated watchdog to cope with the problem of the direct detection of selfish behaviors in multi-channel environments.

Acknowledgments

This work was partially supported by MIUR in the framework of the PRIN SESAME project.

7. REFERENCES

[1] P. Antoniadis, B. Le Grand, A. Satsiou, L. Tassiulas, R.L. Aguiar, J.P. Barraca, and S. Sargento. Community building over neighborhood wireless mesh networks. *IEEE Technology and Society*, 27(1), 2008.

[2] N. Nandiraju, D. Nandiraju, L. Santhanam, B. He, J. Wang, and DP Agrawal. Wireless mesh networks: Current challenges and future directions of web-in-the-sky. *IEEE Wireless Communications*, 14(4):79–89, 2007.

[3] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[4] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. *Proc. of the 6th annual International Conference on Mobile Computing and Networking*, pages 255–265, 2000.

[5] Z. Zhang, F. Näıt-Abdesselam, PH Pin, and X. Lin. RADAR: a ReputAtion-based scheme for Detecting Anomalous nodes in wiReless mesh networks. *IEEE Wireless Communications and Networking Conference, WCNC*, pages 2621–2626, 2008.

[6] C. Zouridaki, B.L. Mark, M. Hejmo, and R.K. Thomas. E-Hermes: A robust cooperative trust establishment scheme for mobile ad hoc networks. *Ad Hoc Networks*, 7(6):1156–1168, 2009.

[7] Y.L. Sun, W. Yu, Z. Han, and K.J.R. Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305, 2006.

[8] D. Quercia, S. Hailes, and L. Capra. B-trust: Bayesian trust framework for pervasive computing. *Lecture Notes in Computer Science*, 3986:298–312, 2006.

[9] A. Jøsang and R. Ismail. The beta reputation system. *Proc. of the 15th Bled Electronic Commerce Conference*, pages 324–337, 2002.

[10] Saurabh Ganeriwal, Laura K. Balzano, and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):1–37, 2008.

[11] J. Luo, X. Liu, and M. Fan. A trust model based on fuzzy recommendation for mobile ad-hoc networks. *Computer Networks*, article in press, 2009.

[12] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos. Stimulating participation in wireless community networks. *IEEE INFOCOM*, pages 1–13, April 2006.

[13] S. Zhong, J. Chen, and Y.R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. *IEEE INFOCOM*, 3(30):1987–1997, 2003.

[14] L. Buttyan and J.P. Hubaux. Enforcing service availability in mobile ad-hoc wans. *Proc. of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 87–96, 2000.

[15] F. Martignon, S. Paris, and A. Capone. Design and Implementation of MobiSEC: a Complete Security Architecture for Wireless Mesh Networks. *Elsevier Computer Networks*, article in press, April 2009.

[16] J. Kim and S. Bahk. Design of certification authority using secret redistribution and multicast routing in wireless mesh networks. *Computer Networks*, 53(1):98–109, 2009.

[17] T. Moreton and A. Twigg. Enforcing collaboration in peer-to-peer routing services. *Lecture notes in computer science*, 2692/2003:255–270, 2003.

[18] L. Bononi and C. Tacconi. Intrusion Detection for Secure Clustering and Routing in Mobile Multi-hop Wireless Networks. *International Journal of Information Security*, 6(6):379–392, 2007.