Institut Eurécom
2229 Route des Crêtes - BP 193
06904 Sophia-Antipolis, France

Research Report N° RR-02-062

CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks.

Pietro Michiardi – Refik Molva
December 2001

Phone:
+33.4.93.00.26.45
+33.4.93.00.26.12

e-Mail:
Piero.Michiardi@eurecom.fr
Refik.Molva@eurecom.fr

Abstract. Countermeasures for node misbehavior and selfishness are mandatory requirements in MANET. Selfishness that causes lack of node activity cannot be solved by classical security means that aim at verifying the correctness and integrity of an operation. We suggest a generic mechanism based on reputation to enforce cooperation among the nodes of a MANET to prevent selfish behavior. Each network entity keeps track of other entities' collaboration using a technique called reputation. The reputation is calculated based on various types of information on each entity's rate of collaboration. Since there is no incentive for a node to maliciously spread negative information about other nodes, simple denial of service attacks using the collaboration technique itself are prevented. The generic mechanism can be smoothly extended to basic network functions with little impact on existing protocols.

Keywords. Security, Mobile Ad hoc Networks, Attacks.

Table of Contents.

CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks

1 INTRODUCTION

A simulation study presented in [1] showed that the performance of MANET severely degrades in face of simple node misbehavior. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks, those functions are carried out by all available nodes. This very difference is at the core of the increased sensitivity to node misbehavior in ad hoc networks.

If a priori trust relationship exists between the nodes of an ad hoc network, entity authentication can be sufficient to assure the correct execution of critical network functions. A priori trust can only exist in a few special scenarios like military networks and requires tamper-proof hardware for the implementation of critical functions. Entity authentication in a large network on the other hand raises key management requirements.

If tamper-proof hardware and strong authentication infrastructure are not available, the reliability of basic functions like routing can be endangered by any node of an ad hoc network. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also requires that each node performs a fair share of the functions. No classical security mechanism can help counter a misbehaving node in this context.

Apart from special cases whereby an a priori trust exists in all nodes, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions. Essential network operations assuring basic connectivity can be heavily jeopardized by nodes that do not properly execute their share of the network operations like routing, packet forwarding, name-to-address mapping, etc. Node misbehavior that affects these operations may range from simple selfishness or lack of collaboration due to the need for power saving to active attacks aiming at denial of service (DoS) and subversion of traffic. *Selfish nodes* use the network but do not cooperate, saving battery life for their own communications: they do not intend to directly damage other nodes. *Malicious nodes*, on the other hand, aim at damaging other nodes by causing network outage by partitioning while saving battery life is not a priority.

Because of their increased vulnerability, ad hoc networks should take into account security problems as a basic requirement regardless of the application scenarios and countermeasures must be integrated with basic networking mechanisms at the early stages of their design. Security mechanisms that solely enforce the correctness or integrity of network operations would thus not be sufficient in MANET. A basic requirement for keeping the network operational is to enforce ad hoc nodes' contribution to network operations despite the conflicting tendency of each node towards selfishness as motivated by the scarcity of node power.

We propose a mechanism called CORE to enforce node cooperation based on a collaborative monitoring technique. CORE is suggested as a generic mechanism that can be integrated with any network function like packet forwarding, route discovery, network management, and location management. Each network entity in CORE keeps track of other entities' collaboration using a technique called reputation. The reputation metric is computed based on data monitored by the local entity and some information provided by other nodes involved in each operation. An interesting feature of the CORE mechanism is that denial of service attacks based on malicious broadcasting of negative ratings for legitimate nodes are prevented.

The remainder of the paper is organized as follows: section 2 introduces the basic reputation concept underlying the CORE mechanism, the generic CORE mechanism presented in section 3 is

then illustrated with the applications of this mechanism to packet forwarding and routing functions in section 4.

2 THE REPUTATION CONCEPT

In our scheme, MANET nodes can be thought of as members of a community (or subjects) that share a common resource. The key to solve problems related to node misbehavior derives from the strong binding between the utilization of a common resource and the cooperative behavior of the members of the community. Thus, all members of a community that share resources have to contribute to the community life in order to be entitled to use those resources. However, the members of a community are often unrelated to each other and have no information on one another's behavior. We believe that reputation is a good measure of someone's contribution to common network operations. Indeed, reputation is usually defined as the amount of trust inspired by a particular member of a community in a specific setting or domain of interest. Members that have a good reputation, because they helpfully contribute to the community life, can use the resources while members with a bad reputation, because they refused to cooperate, are gradually excluded from the community.

The approach presented in this section is used as a basis for the security mechanism that solves the problems due to misbehaving nodes by incorporating a reputation mechanism that provides an automatic method for the social mechanisms of reputation. Furthermore the formulae presented in the following sections are conceived in order to minimize problems due to false detection of a nodes' misbehavior. As an example, disadvantaged nodes that are inherently selfish due to their precarious energy conditions shouldn't be excluded from the network using the same basis as for malicious nodes: this is done with an accurate evaluation of the reputation value that takes into account a sporadic misbehavior.

2.1 Definitions

This section presents the three types of reputation used in our scheme and shows how they are combined. Reputation is formed and updated along time through direct observations and through information provided by other members of the community. Furthermore, we take the stance that reputation is compositional: the overall opinion on an entity that belongs to the community is obtained as a result of the combination of different type of evaluations. We define a subjective reputation, an indirect reputation and a functional reputation.

2.1.1 Subjective Reputation

We use the term subjective reputation to talk about the reputation calculated directly from a subject's observation. A subjective reputation at time t from subject s_i point of view is calculated using a weighted mean of the observations' rating factors, giving more relevance to the past observations.

The reason why more relevance is given to past observations is that a sporadic misbehavior in recent observations should have a minimal influence on the evaluation of the final reputation value: as a result, it is possible to avoid false detections due to link breaks and to take into account the possibility of a localized misbehavior caused by disadvantaged nodes.

The general formula to calculate a subjective reputation is:

$$r_{s_i}^{t}\left(s_j \middle| f\right) = \sum \rho(t,t_k) \cdot \sigma_k$$

where $r_{s_i}^{t}\left(s_j \middle| f\right)$ stands for the subjective reputation value calculated at time t by subject s_i on subject s_j with respect to the function f.

$\rho(t,t_k)$ is a time dependent function that gives higher relevance to past values of σ_k.

σ_k represents the rating factor given to the *k-th* observation: we use a scale that goes from -1 for a negative impression (meaning that the observed result doesn't match with the expected result) to +1 for a positive impression (i.e. when the observed and the expected results coincides).
When the number or the quality of observations collected since time *t* are not sufficient, the final value of the subjective reputation takes the 0 value, which is used for a neutral impression.
Finally, given that $\sigma_k \in [-1,1]$ and that $\rho(t,t_k)$ is a normalized value, also $r_{s_i}^t(s_j|f) \in [-1,1]$.

Note also that the set $\{s_j\}$ is restricted to the set of the neighbors of subject s_i. We use the term neighbor to refer to a subject that is within wireless transmission range of another subject.

2.1.2 Indirect Reputation

In our scheme, the subjective reputation is evaluated only considering the direct interaction between a subject and its neighbors. With the introduction of the indirect reputation measure we add the possibility to reflect in our model a characteristic of complex societies: the final value given to the reputation of a subject is influenced also by information provided by other members of the community.
In the reminder of the paper, $ir_{s_i}^t(s_j|f)$ denotes the indirect reputation of subject s_j collected by s_i at time *t* for the function *f*.
The information collected through indirect reputation can take only positive values: denial of service attacks based on malicious broadcasting of negative ratings for legitimate nodes are thus prevented.

2.1.3 Functional Reputation

We use the term functional reputation to talk about the subjective and indirect reputation calculated with respect to different functions *f*. With the introduction of this last type of reputation in our model we add the possibility to calculate a global value of a subject's reputation that takes into account different observation/evaluation criteria. As an example, a subject s_i can evaluate the subjective reputation $r_{s_i}^t(s_j|f(\text{packet forwarding}))$ of subject s_j with respect to the packet forwarding function and the subjective reputation $r_{s_i}^t(s_j|f(\text{routing}))$ with respect to the routing function and combine them using different weights to obtain a global reputation value on subject s_j.

2.1.4 Combination of reputation information for multiple functions

Reputation information is combined using the following formula:

$$r_{s_i}^t(s_j) = \sum_k w_k \cdot \left\{ r_{s_i}^t(s_j|f_k) + ir_{s_i}^t(s_j|f_k) \right\}$$

where w_k represents the weight associated to the functional reputation value.
$r_{s_i}^t(s_j)$ represents the global reputation value that is evaluated in every node: it is the aggregate reputation definition.

The choice of the weights w_k used to evaluate the global reputation has to be accurate because it can affect the overall system robustness. The simulation study carried out in [1] pointed out that even if the enforcement of the execution of both the packet forwarding function and the routing function are mandatory, the former has an important impact on the global performances compared to the latter. This is why a good choice for w_k would emphasize the correctness of the packet forwarding function when evaluating the overall reputation for a node.
Besides the global reputation value, it is important to know how reliable is that value. Although there are a lot of elements that can be taken into account to calculate how reliable a global reputation is, we propose two of them: the number of evaluations used to calculate the final reputation value and its variance. This approach is similar to that used in the Sporas system [9].

2.1.5 Validation mechanism

Each type of reputation is obtained as a combination of different observations made by a subject over another subject with respect to a defined function *f* every observation is related to the correct execution of *f*. It is necessary to define a validation mechanism (based on feed back information) that compares the observed results and the expected results and checks whether they coincides or not. If the objectives have been reached (i.e. observed and expected results coincides) then the rating factor σ_k associated to the *k-th* observation will be positive, while if the observation shows that the expected results are not reached (i.e. the function f has not been correctly executed) then the rating factor will be negative.

More details on the validation mechanism will be given in the section 3.1.3 where we consider a possible implementation: the watchdog mechanism.

3 THE CORE SCHEME

This section presents the CORE scheme in details, starting from the definition of the components that participate to the collaborative reputation mechanism and concluding with the description of the complete process in which the different parts are involved.

3.1 Components

3.1.1 Network entity

The network entity corresponds to a mobile node. Each entity s_i is enriched with a set of Reputation Tables (RT) and a watchdog mechanism (WD). The RT and the WD together constitute the basis of the collaborative reputation mechanism presented in this paper. These two components allow each entity to observe and classify each other entity that gets involved in a request/reply process, reflecting the cooperative behavior of the involved parts. The classification of the entities based on their behavior is then used to enforce the strong binding between the cooperative behavior of a subject and the utilization of the common resources made available by all the other entities of the network.

We use the notation *requestor* when referring to a network entity asking for the execution of a function *f* and the notation *provider* when referring to any entity supposed to correctly execute *f*. We also use the notation *trusted entity* when referring to a network entity with a positive value of reputation.

3.1.2 Reputation Table

The Reputation Table (RT) is defined as a data structure stored in each network entity. Each row of the table includes the reputation data pertaining to a node. Each row consists of four entries: the unique identifier of the entity, a collection of recent subjective observations made on that entity's behavior, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated for a predefined function.

Each network entity has one RT for each function that has to be monitored. Finally, a global RT is used to combine the different values of reputation calculated for different functions, as explained in section 2.1.

The mechanism used to update and distribute the RTs will be explained in section 3.2.

3.1.3 The Watchdog mechanism

The watchdog (WD) mechanism implements the validation phase depicted in section 2.1 and it is used to detect misbehaving nodes. Every time a network entity ($s_{i,m}$, monitoring entity) needs to monitor the correct execution of a function implemented in a neighboring entity ($s_{j,o}$, observed entity), it triggers a WD specific to that function (*f*). The WD stores the expected result $e_r(f)$ in a temporary buffer in $s_{i,m}$ and verifies if the observed result $o_r(f)$ and $e_r(f)$ match. If the monitored

function is executed properly then the WD removes from the buffer the entry corresponding to the $s_{j,o}$,$e_r(f)$ couple and enters in an idle status, waiting for the next function to observe. On the other hand, if the function is not correctly executed or if the couple $s_{j,o}$,$e_r(f)$ remains in the buffer for more than a certain time out, a negative value to the observation rating factor σ_k is reported to the entry corresponding to $s_{j,o}$ in the RT and a new reputation value for that entity is calculated.
It should be noticed that the term *expected result* corresponds to the correct execution of the function monitored by the WD, which is substantially different from the final result of the execution of the function.

The principles presented in this section lack to show the limitations related to a real implementation of the WD mechanism: more details will be given in section 4.1.

3.2 Protocol

In this section we present the protocol used to support the collaborative reputation mechanism introduced in the previous sections. Each party involved in the protocol corresponds to a network entity, as defined in section 3.1.1.
The CORE scheme involves two types of protocol entities, a *requestor* and one or more *providers*, that are within the wireless transmission range of the *requestor*. The nature of the protocol and the mechanisms on which it relies assure that if a provider refuses to cooperate (i.e. the request is not satisfied), then the CORE scheme will react by decreasing the reputation of the provider, leading to its exclusion if the non-cooperative behavior persists. More details on the effects that a non-cooperative behavior has on an entity's reputation and the mechanism used to exclude the misbehaving entity will be given respectively in section 3.4 and in section 3.5.

For sake of simplicity, the following scenarios are related to the execution of the protocol between a *requestor* and one *provider*.

3.2.1 Protocol execution when no misbehavior is detected

First, the *requestor* asks for the execution of a function *f* to the *provider*. It then activate the WD related to the *provider* for the required *f* and waits for the outcome of the WD within a predefined time out. Since the two parties correctly behave, the outcome of the WD assures that the requested function was correctly executed and the *requestor* disarms the WD.
We suppose that the reply message corresponding to the result of the execution of function *f* includes a list of all the entities that correctly participated to the protocol: the *requestor* uses this indirect information to update its RT and enters in an idle mode.

3.2.2 Protocol execution when misbehavior is detected

As described in the previous scenario, the *requestor* asks for the execution of a function *f* and arms the related WD, waiting for the outcome. Since we suppose that the provider does not cooperate, the outcome of the watchdog will be negative. The *requestor* will then update the entry in the RT corresponding to the misbehaving entity with a negative factor and will enter in an idle mode.

3.2.3 Request made by a misbehaving entity

We describe here the process that any entity receiving a request has to follow. Upon receiving the request for the execution of a function *f* the entity checks the reputation value evaluated for the *requestor* in its global RT. If the reputation value is negative then the entity will not execute the requested function. It has then the choice whether to notify or not the denial of service. A detailed analysis on the best practice will be presented in section 3.5.

3.3 RT updates and distribution

We focus now on the mechanism used to update and distribute reputation information. RTs are updated in two different situations: during the request phase of the protocol and during the reply phase corresponding to the result of the execution of *f*.

In the first case, it is possible to notice that only the subjective reputation value is updated. If the outcome of the WD shows that the *provider* did not cooperate, a negative rating factor will be assigned to the observation and consequently the reputation related to the misbehaving entity will decrease. If no misbehavior is detected, the RTs are not updated.

In the second case, only the indirect reputation value is updated. We suppose that the reply message contains a list of all the entities that correctly behaved: the indirect reputation will be positive and consequently the reputation related to the cooperating entities will increase.

The reason why only positive rating factors can be distributed among the entities while the negative rating factors are evaluated locally derives from a possible attack to the protocol. If negative factors could be spread around, it would be simple for a misbehaving entity to distribute false information about other entities in order to initiate a denial of service (DoS) attack. The protocol presented in this paper allows only the distribution of positive rating factors: if we suppose a scenario where collusion between misbehaving entities is impossible, then there would be no advantage for a misbehaving entity to distribute positive rating factors to other unknown entities. Furthermore, reputation information is distributed and updated only during the reply phase avoiding a indiscriminate broadcast of bogus information.

In a possible variation of the protocol, both positive and negative rating factors could be broadcasted during the reply phase. The distribution of indirect reputation information would however only be limited to adjacent entities. Furthermore, as described in section 2.1, the influence of any indirect information on the final value of the reputation is mitigated by giving more relevance to the information collected from trusted entities.

Reputation values calculated for each entry of the RT are not constant: if the reputation value is positive then it is decremented along time. The reason why we decided to decrement positive reputation values comes from a possible attack to the CORE scheme: if a network entity enters in an idle status for most of the time except when it has to communicate, its reputation has to be decreased, even if during the active time it cooperates to the network operation. Reputation is decreased until it reaches a null value, which corresponds to a neutral behavior meaning that it is nor a misbehaving entity neither a cooperating entity.

A preliminary analysis of the effects of the two approaches on the RTs will be given in section 3.4 but we believe that a simulation-based study is necessary in order to determine the optimal setting of protocol parameters.

3.4 Effects of misbehavior on the Reputation Table

Figure 1 shows a typical scenario composed of network entities that communicate with one another. The dotted circle around each entity represents the wireless transmission range of each node. Any entity colored in dark represents a misbehaving entity.

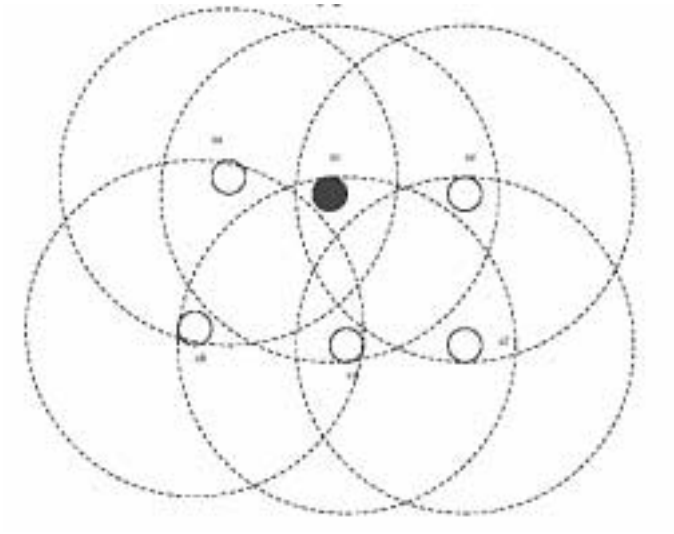

Figure 1. Typical scenario.

We analyze in this section the effects of an entity's misbehavior with respect to a generic function *f*. It should be noted that the RT update process and the calculation of a reputation value depend on the function that has to be executed. Sections 4.3 and 4.4 presents an application of the CORE scheme when the function is respectively the DSR Route Discovery function and the Packet Forwarding function in a mobile ad hoc network.

Referring to the first approach proposed in section 3.3, the non-cooperative behavior of s_c can be detected by s_a, s_d and s_e. The other entities that belong to the network will not be informed about s_c's misbehavior.

On the other hand, the second approach exposed in section 3.3 would allow the distribution of indirect information to adjacent nodes. Entities s_b and s_f will be informed about s_c's misbehavior, both from s_a and s_d and from s_e and s_d respectively. As a result, cooperation will be enforced by a larger number of entities and the consequences of an entity misbehavior will more significant.

3.5 Cooperation Enforcement

This section describes how reputation information is used to enforce cooperation between entities. Reputation is directly related to the cooperative behavior of an entity: if the reputation value is negative then the entity is classified as a misbehaving entity while if the reputation value is positive then the entity is tagged as a trusted entity. The execution of a function requested by any *requestor* is conditioned by the corresponding reputation value stored in the global RT of the *provider*: when this reputation value is negative then the provider will deny the execution of the requested operation.
There is no advantage for an entity to misbehave because any resource utilization will be forbidden.

Reputation is hard to build because positive rating factors are acquired only in the reply message which contains the list of all the network entities that cooperated to obtain of the final result of the requested function. On the other hand, negative rating factors are attributed every time the outcome of the WD is negative. Even if reputation is not linearly decreased for every negative rating factor in order to avoid false evaluations (e.g. apparent misbehavior due to link breaks), a persistent non-cooperative behavior compromises normal resource utilization leading to the exclusion of the misbehaving entity from the network.

4 APPLICATIONS

4.1 Background and assumptions

This section outlines the assumptions that were made regarding the properties of the physical and network layer of the MANET. Throughout this paper we assume bi-directional communication symmetry on every link between the nodes. This means that if a node B is capable of receiving a message from a node A at time t, then node A could instead have received a message from node B at time t. Furthermore the routing protocol that has been used as a basis for the study of the CORE scheme is the Dynamic Source Routing (DSR) protocol.

This paper addresses MANET with a low node density. We consider each node as being part of a *zone*. However, it is not the aim of this paper to present a mechanism to divide a MANET in zones: we take the stance that a similar mechanism used to organize MANET for hybrid routing can be used to define a zone.

In addition, we assume wireless interfaces that support promiscuous mode operation. Promiscuous mode means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A. The watchdog technique presented in section 3.1.3 relies on the promiscuous mode operation and has some weaknesses that have been presented in [2]. WD's weaknesses are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping. The analysis of WD's weaknesses carried out by the authors is complete and detailed and we suggest the interested reader to refer to [2].

4.2 Node misbehavior model

The node misbehavior model used in this paper take inspiration from the threats presented in [1]. The research presented in [1] pointed out two types of misbehavior: a selfish behavior and malicious behavior. The protocol presented in this paper focuses on the node selfishness problem.

4.3 Application of CORE to the DSR Route Discocery function

Route discovery allows any node in the ad hoc network to dynamically discover a route to any node in the ad hoc network, whether directly reachable within wireless transmission range or reachable through one or more intermediate network hops through other nodes. A node initiating a route discovery broadcasts a route *request* message which may be received by those nodes within wireless transmission range of it. When any node receives a route request message it processes the request and if the target of the request is unknown it appends the node's own address to the route record in the route request packet and re-broadcast the request. If the route discovery is successful the initiating node receives a route *reply* message listing a sequence of network hops through which it may reach the target.

As described in section 3.2, the CORE scheme involves a *requestor* and one or more *providers* that are within the wireless transmission range of the *requestor*. The CORE protocol can be thought of as a layer on top of the DSR protocol, and the function *f* that has to be monitored corresponds to the Route Discovery function of the DSR protocol. The WD mechanism is able to detect any misbehaving node that does not participate to the Route Discovery phase of the protocol and the evaluation of the reputation value reflects any node misbehavior. Node misbehavior is detected in the request phase of the Route Discovery function while the reply phase informs the initiator and the intermediate nodes on the identity of the network entities that participated to the Route Discovery phase: reputation value is updated to reflect the positive rating factors assigned to the cooperating nodes.

Every node stores a set of RTs that are used to classify other nodes of the network: route requests originating from nodes classified as cooperating entities will be served properly whereas routing service will be denied to route requests issued by misbehaving nodes. Only a cooperative behavior allows an entity to change its reputation value from negative to positive: nodes are stimulated to participate to the Route Discovery function if they want to be served when they need to communicate.

4.4 The CORE scheme applied to the Packet Forwarding function

Similarly, the CORE scheme can be used to monitor the Packet Forwarding (PF) function. Once a node has obtained a valid route to the destination through the DSR Route Discovery function, it can start sending data packet to its target. Each network entity belonging to the path from the source to the destination has to perform the PF function in order transfer the data packets. The WD mechanism can be used to detect any misbehaving nodes that refuse to cooperate to the PF and the evaluation of the reputation value reflects any node misbehavior.
As opposed to the Route Discovery function, the PF function does not offer separate operations that can be qualified as request and reply phases. However, if an acknowledgment (ACK) packet can be included in the original data transfer protocol for the purpose of security, the transfer of the data packet can be thought of as the request phase while the transfer of ACK can be considered as the reply phase.
As described in section 4.3 any node misbehavior is detected in the request phase of the PF function while the reply phase informs the initiator and the intermediate nodes on the identity of the network entities that participated to the PF: reputation value is updated to reflect the positive rating factors assigned to the cooperating nodes.

Every node stores a set of RTs that are used to classify other nodes of the network with respect to the PF function. The execution of the PF function is granted for any node classified as a cooperating entity while it is denied for misbehaving nodes. Only a cooperative behavior allows an entity to change its reputation value from negative to positive: nodes are stimulated to participate to the PF function if they want their own data packet to be forwarded to the destination.

5 RELATED WORK

The area of ad hoc networking has been receiving increasing attention among researchers in recent years and a variety of routing protocols targeted specifically at the ad hoc networking environment have been proposed. However, very few researchers focus on the selfishness problem in MANET and existing work in this area is still in its infancy.

In [2], the authors consider the case in which some misbehaving nodes agree to forward packets but fail to do so. In order to solve this problem, they propose two mechanisms: a watchdog, in charge of identifying the misbehaving nodes, and a pathrater, in charge of defining the best route circumventing these nodes. The paper shows that these two mechanisms make it possible to maintain the total throughput of the network at an acceptable level, even in the presence of a high amount of misbehaving nodes (e.g., 40%). However, the operation of the watchdog is based on an assumption which is not always true (as reckoned by the authors): the promiscuous mode of the wireless interface. Another problem is that the selfishness of the nodes does not seem to be castigated; on the contrary, by the combination of the watchdog and the pathrater, the misbehaving nodes will not be bothered by the transit traffic, while still enjoying the possibility to generate and to receive traffic.

CORE differs from the watchdog-pathrater scheme as follows:

in CORE misbehaving nodes are stimulated to contribute to the network operations in order to be able to use network services, the pathrater mechanism helps a legitimate user to avoid using misbehaving nodes;

CORE is a generic mechanism that can be integrated with several network and application layer functions whereas the watchdog-pathrater scheme is specifically designed for routing;

unlike the pathrater technique the reputation mechanism in CORE does not allow a node to distribute negative ratings about other nodes, so unlike the pathrater technique, CORE can resist to simple denial of service attacks that use the security mechanism itself.

In [7], the authors present two important issues targeted specifically at the ad hoc networking environment: first, end-users must be given some incentive to cooperate to the network operation (especially to relay packets belonging to other nodes); second, end-users must be discouraged from overloading the network. The solution presented in their paper consists in the introduction of a virtual currency (that they call Nuglets) used in every transaction. Two different models are described: the Packet Purse Model and the Packet Trade Model. In the Packet Purse Model each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding service. The advantage of this approach is that it discourages users from flooding the network but the drawback is that the source needs to know exactly how many nuglets it has to include in the packet it sends. In the Packet Trade Model each packet is traded for nuglets by the intermediate nodes: each intermediate node buys the packet from the previous node on the path. Thus, the destination has to pay for the packet. The direct advantage of this approach is that the source does not need to know how many nuglets need to be loaded into the packet. On the other hand, since the packet generation is not charged, malicious flooding of the network cannot be prevented. There are some further issues that have to be solved: concerning the Packet Purse Model, the intermediate nodes are able to take out more nuglets than they are supposed to; concerning the Packet Trade Model, the intermediate nodes are able to deny the forwarding service after taking out nuglets from a packet.

6 FUTURE WORK

The security approach presented in this paper will be completed with an accurate analysis and classification of denial of service attacks specific to the ad hoc networks environment. Indeed, in this paper we considered only selfishness as a specific issue to address: selfish nodes, however, do not intend to directly damage other nodes while the misbehavior is due to their need to save battery life for their own communications. Our ongoing research is evaluating the robustness of the proposed scheme when we consider also malicious nodes that aim at damaging other nodes. In this case, active denial of service attacks can be performed by malicious nodes and our work focus on the definition of other possible attacks.

Furthermore, we focus also on the definition of a formal method, based on the game theory, to analytically prove the robustness of our scheme: we expect to demonstrate that the security mechanism exposed in the paper is compliant to our security objectives.

An in-depth analysis of our security scheme is ongoing using our simulation environment. Our goal is to implement a wide choice of attacks using the QualNet network simulator: we enhanced our software by adding passive denial of service attacks perpetrated on the packet forwarding function and the routing function and we plan to add new features including active denial of service attacks and traffic subversion. We also aim at extending our misbehavior model in order to consider eventual collusions between malicious entities.

The analysis of the simulation results is based on an appropriate metric we defined in order to give emphasis to the robustness of a generic security scheme with respect to the percentage of misbehaving nodes present in the network. We also plan to analyze the performances of our

mechanism with respect to node mobility and node density: we believe that network characteristics can be used as trigger signals for the fine tuning of our scheme.

7 CONCLUSIONS

The area of ad hoc network security has been receiving increasing attention among researchers in recent years. However, little has been done so far in terms of the definition of security needs specific to different types of scenario that can be defined for ad hoc networks. We introduced a fundamental distinction between ad hoc networks where an a priori trust relationship exists between the nodes, provided as an example by a common authority, and ad hoc networks where there is no shared a priori trust between the mobile nodes.

Our research is focused on MANET where there is a lack of a priori trust relationship between mobile nodes. Countermeasures against node misbehavior in general and denial of service attacks in particular is our very first concern. In this paper we suggested a generic mechanism based on reputation to enforce cooperation among the nodes of a MANET and to prevent passive denial of service attacks due to node selfishness. This mechanism can be smoothly extended to basic network functions with little impact on existing protocols.

References

[1] P. Michiardi, R. Molva. Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks. European Wireless Conference, 2002.

[2] S. Marti, T. Giuli, K. Lai, and M. Baker. *Mitigating routing misbehavior in mobile ad hoc networks*. In Proceedings of MOBICOM, 2000.

[3] The Terminodes Project. www.terminodes.org.

[4] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J-P. Hubaux, and J-Y. Le Boudec. *Self-organization in mobile ad hoc networks: The approach of Terminodes*. IEEE Communications Magazine, June 2001.

[5] L. Buttyan and J-P. Hubaux. *Enforcing service availability in mobile ad hoc networks*. In proceedings of MobiHOC, 2000.

[6] J.-P. Hubaux, T. Gross, J.-Y. Le Boudec, and M. Vetterli. *Toward self-organized mobile ad hoc networks: The Terminodes Project*. IEEE Communications Magazine, January 2001.

[7] L. Buttyan and J.-P. Hubaux. *Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks*. Technical Report DSC/2001/001, Swiss Federal Institute of Technology -- Lausanne, 2001.

[8] L. Zhou and Z. Haas. *Securing ad hoc networks*. IEEE Network, 13(6):24--30, November/December 1999.

[9] G. Zacharia. Collaborative Reputation Mechanisms for online communities. Master's thesis, MIT, September 1999.