CoCoWa: A Collaborative Contact-based Watchdog for Detecting Selfish Nodes

Enrique Hernández-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni
Departamento de Informática de Sistemas y Computadores. Universitat Politècnica de València. Spain.
emails: ehernandez@disca.upv.es, mdserrat@upvnet.upv.es, {jucano, calafate, pmanzoni}@disca.upv.es

***Abstract*—Mobile Ad-hoc Networks (MANETs) assume that mobile nodes voluntary cooperate in order to work properly. This cooperation is a cost-intensive activity and some nodes can refuse to cooperate, leading to a selfish node behaviour. Thus, the overall network performance could be seriously affected. The use of watchdogs is a well-known mechanism to detect selfish nodes. However, the detection process performed by watchdogs can fail, generating false positives and false negatives that can induce to wrong operations. Moreover, relying on local watchdogs alone can lead to poor performance when detecting selfish nodes, in term of precision and speed. This is specially important on networks with sporadic contacts, such as Delay Tolerant Networks (DTNs), where sometimes watchdogs lack of enough time or information to detect the selfish nodes. Thus, we propose CoCoWa (Collaborative Contact-based Watchdog) as a collaborative approach based on the diffusion of local selfish nodes awareness when a contact occurs, so that information about selfish nodes is quickly propagated. As shown in the paper, this collaborative approach reduces the time and increases the precision when detecting selfish nodes.**

***Index Terms*—Wireless networks, MANETs, Opportunistic and Delay Tolerant Networks, Selfish Nodes, Performance Evaluation.**

I. Introduction

Cooperative networking is currently receiving significant attention as an emerging network design strategy for future mobile wireless networks. Successful cooperative networking can prompt the development of advanced wireless networks to cost-effectively provide services and applications in contexts such as vehicular ad hoc networks (VANETs) or mobile social networks. Two of the basic technologies that are considered as the core for these types of networks are Mobile Ad-Hoc Networks (MANETs) and Opportunistic and Delay Tolerant Networks (DTNs).

The cooperation on these networks is usually contact-based. Mobile nodes can directly communicate with each other if a contact occurs (that is, if they are within communication range). Supporting this cooperation is a cost intensive activity for mobile nodes. Thus, in the real world, nodes could have a selfish behaviour, being unwilling to forward packets for others. Selfishness means that some nodes refuse to forward other nodes' packets to save their own resources.

The literature provides two main strategies to deal with selfish behaviour: a) motivation or incentive based approaches, and b) detection and exclusion. The first approach, tries to motivate nodes to actively participate in the forwarding activities. These approaches are usually based on virtual currency and/or game theory models [4], [5], [9], [36]. The detection and exclusion approach is a straight-forward way to cope with selfish nodes and several solutions have been presented [3], [14], [19], [22]–[25], [28], [34]. In CoCoWa, we do not attempt to implement any strategy to exclude selfish nodes or to incentivize their participation; instead, we focus on the detection of selfish nodes.

The impact of node selfishness on MANETs has been studied in [30]–[32]. In [32] it is shown that when no selfishness prevention mechanism is present, the packet delivery rates become seriously degraded, from a rate of 80% when the selfish node ratio is 0, to 30% when the selfish node ratio is 50%. The survey [31] shows similar results: the number of packet losses is increased by 500% when the selfish node ratio increases from 0% to 40%. A more detailed study [30] shows that a moderate concentration of node selfishness (starting from a 20% level) has a huge impact on the overall performance of MANETs, such as the average hop count, the number of packets dropped, the offered throughput, and the probability of reachability. In DTNs, selfish nodes can seriously degrade the performance of packet transmission. For example, in two-hop relay schemes, if a packet is transmitted to a selfish node, the packet is not re-transmitted, therefore being lost.

Therefore, detecting such nodes quickly and accurately is essential for the overall performance of the network. Previous works have demonstrated that watchdogs are appropriate mechanisms to detect misbehaving and selfish nodes. Essentially, watchdog systems overhear wireless traffic and analyse it to decide whether neighbour nodes are behaving in a selfish manner [16]. When the watchdog detects a selfish node it is marked as a *positive detection* (or a *negative detection*, if it is detected as a non selfish node). Nevertheless, watchdogs can fail on this detection, generating *false positives* and *false negatives* that seriously degrade the behaviour of the system.

Another source of problems for cooperative approaches is the presence of colluding or malicious nodes. In this case, the effect can even be more harmful, since these nodes try to intentionally disturb the correct behaviour of the network. For example, one harmful malicious node can be lying about the status of other nodes, producing a fast diffusion of false negatives or false positives. Malicious nodes are hard to detect using watchdogs, as they can intentionally participate in network communication with the only goal to hide their behaviour from the network. Thus, since we assume that these nodes may be present on the network, evaluating their influence becomes a very relevant matter.

This paper introduces CoCoWa (*Collaborative Contact-based Watchdog*) as a new scheme for detecting selfish nodes that combines local watchdog detections and the dissemination of this information on the network. If one node has previously detected a selfish node it can transmit this information to other nodes when a contact occurs. This way, nodes have second hand information about the selfish nodes in the network. The goal of our approach is to reduce the detection time and to improve the precision by reducing the effect of both false negatives and false positives. Although some of the aforementioned papers (such as [3], [28]) introduced some degree of collaboration on their watchdog schemes, the diffusion is very costly since they are based on periodic message dissemination.

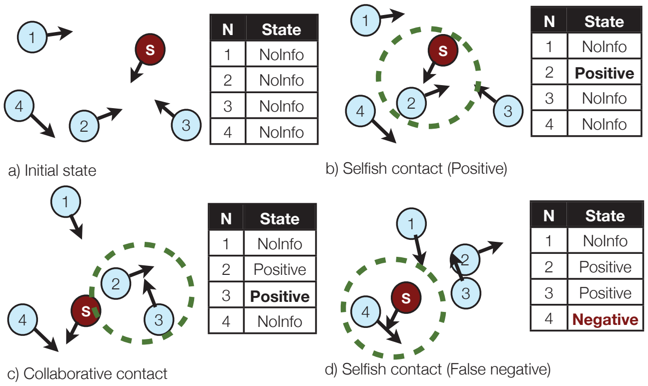
The diffusion of information about positive or negative detections of selfish nodes introduces several issues about the reputation of the neighbour nodes. The first issue is the consolidation of information, that is, the trust about neighbour's positive and negative detections, specially when it does not match with the local watchdog detection. Another issue is the case of malicious nodes. Thus, this paper extends our previous approaches [12], [13] to also cope with malicious nodes using a reputation scheme.

In order to evaluate the efficiency of CoCoWa we first introduce an analytical performance model. We model the network as a Continuous Time Markov Chain (CTMC) and derive expressions for obtaining the time and overhead (cost) of detection of selfish nodes under the influence of false positives, false negatives and malicious nodes. In general, the analytical evaluation shows a significant reduction of the detection time of selfish nodes with a reduced overhead when comparing CoCoWa against a traditional watchdog. The impact of false negatives and false positives is also greatly reduced. Finally, the pernicious effect of malicious nodes can be reduced using the reputation detection scheme. We also evaluate CoCoWa with real mobility scenarios using well known human and vehicular mobility traces. These experimental results confirm that our approach is very efficient.

The rest of the paper is organised as follows. We first introduce the architecture of CoCoWa in section II. Section III discusses the characterisation of contact occurrence. Then, section IV presents a performance model for evaluating our approach. Section V presents the evaluation of CoCoWa in terms of detection time and overhead using the analytical model. The CoCoWa approach is also experimentally evaluated using real mobility traces in section VI. After presenting and evaluating our proposal we present some related work in section VII. Finally, section VIII presents the concluding remarks.

II. Architecture overview

A selfish node usually denies packet forwarding in order to save its own resources. This behaviour implies that a selfish node neither participates in routing nor relays data packets [21]. A common technique to detect this selfish behaviour is network monitoring using local watchdogs. A node's watchdog consists on overhearing the packets transmitted and received by its neighbours in order to detect anomalies, such as the ratio between *packets received* to *packets being re-transmitted* [15]. By using this technique, the local watchdog can generate a positive (or negative) detection in case the node is acting selfishly (or not).



a) Initial state

N	State
1	NoInfo
2	NoInfo
3	NoInfo
4	NoInfo

b) Selfish contact (Positive)

N	State
1	NoInfo
2	**Positive**
3	NoInfo
4	NoInfo

c) Collaborative contact

N	State
1	NoInfo
2	Positive
3	**Positive**
4	NoInfo

d) Selfish contact (False negative)

N	State
1	NoInfo
2	Positive
3	Positive
4	**Negative**

Fig. 1: An example of how CoCoWa works. a) Initially all nodes have no information about the selfish node. b) Node 2 detects the selfish node using its own watchdog. c) Node 2 contacts with node 3 and it transmits the positive about the selfish node. d) The local watchdog of Node 4 fails to detect the selfish node and it generates a negative detection (a false negative).

An example of how CoCoWa works is outlined in figure 1. It is based on the combination of a local watchdog and the diffusion of information when contacts between pairs of nodes occurs. A contact is defined as an opportunity of transmission between a pair of nodes (that is, two nodes have enough time to communicate between them). Assuming that there is only one selfish node, the figure shows how initially no node has information about the selfish node. When a node detects a selfish node using its watchdog, it is marked as a *positive*, and if it is detected as a non selfish node, it is marked as a *negative*. Later on, when this node contacts another node, it *can* transmit this information to it; so, from that moment on, both nodes store information about this positive (or negative) detections. Therefore, a node can become aware about selfish nodes directly (using its watchdog) or indirectly, through the collaborative transmission of information that is provided by other nodes.

Under this scheme, the uncontrolled diffusion of positive and negative detections can produce the fast diffusion of wrong information, and therefore, a poor network performance. For example, in figure 1, on the last state d), node two and three have a positive detection and node four has a negative detection (a false negative). Now, node one, which has no information about the selfish node, has several possibilities: if it contacts the selfish node it may be able to detect it; if it contacts node two or three it can get a positive detection; but if it contacts node four, it can get a false negative.

Figure 2 shows the functional structure of CoCoWa and we now detail its three main components.

The *Local Watchdog* has two functions: the detection of selfish nodes and the detection of new contacts. The *local watchdog* can generate the following events about neighbour nodes: `PosEvt` (*positive event*) when the watchdog detects a selfish node, `NegEvt` (*negative event*) when the watchdog detects that a node is not selfish, and `NoDetEvt` (*no detection event*) when the watchdog does not have enough information about a node (for example if the contact time is very low or it does not overhear enough messages). The detection of new contacts is based on *neighbourhood* packet overhearing; thus, when the watchdog overhears packets from a new node it is assumed to be a new contact, and so it generates an event to the network information module.

The *Diffusion* module has two functions: the transmission as well as the reception of positive (and negative) detections. A key issue of our approach is the diffusion of information. As the number of selfish nodes is low compared to the total number of nodes, positive detections can always be transmitted with a low overhead. However, transmitting only positive detections has a serious drawback: false positives can be spread over the network very fast. Thus, the transmission of negative detections is necessary to neutralise the effect of these false positives, but sending all known negative detections can be troublesome, producing excessive messaging or the fast diffusion of false negatives. Consequently, we introduce a *negative diffusion factor* γ, that is the ratio of negative detections that are actually transmitted. This value ranges from 0 (no negative detections are transmitted) to 1 (all negative detections are transmitted). We will show in the evaluation section that a low value for the γ factor is enough to neutralise the effect of false positives and false negatives. Finally, when the diffusion module receives a new contact event from the watchdog, it transmits a message including this information to the new neighbour node. When the neighbour node receives a message, it generates an event to the network information module with the list of these positive (and negative) detections.

Updating or consolidating the information is another key issue. This is the function of the *Information Update* module. A node can have the following internal information about other nodes: `NoInfo` state, `Positive` state and `Negative` state. A `NoInfo` state means that it has no information about a node, a `Positive` state means it believes that a node is selfish, and a `Negative` state means it believes that a node is not selfish. A node can have direct information (from the local watchdog) and indirect information (from neighbour nodes). CoCoWa is event driven, so the state of a node is updated when the `PosEvt` or `NegEvt` events are received from the local watchdog and diffusion modules. In particular, these events updates a *reputation* value ρ using the following expression:

$$\rho = \rho + \Delta \quad \Delta = \begin{cases} +\delta & (\texttt{PosEvt}, \text{Local}) \\ +1 & (\texttt{PosEvt}, \text{Indirect}) \\ -\delta & (\texttt{NegEvt}, \text{Local}) \\ -1 & (\texttt{NegEvt}, \text{Indirect}) \end{cases} \quad \delta \geq 1 \tag{1}$$

In general, a `PosEvt` event increments the reputation value while a `NegEvt` event decrements it. Defining θ as a threshold and using the reputation value ρ, the state of the node changes to `Positive` if $\rho \geq \theta$, and to `Negative` if $\rho \leq -\theta$. Otherwise, the state is `NoInfo`. The combination of δ and θ parameters allows a very flexible and dynamic behaviour. First, if $\theta > 1$ and $\delta < \theta$ we need several events in order to change the state. For example, starting from the `NoInfo` state, if $\theta = 2$ and $\delta = 1$, at least a local and an indirect event is needed to change the state, but if $\theta = 1$, only one event is needed. Second, we can give more trust to the local watchdog or to indirect information. For example, a value of $\delta = 2$ and $\theta = 3$, means that we need one local event and one indirect event, or three indirect events, to change the state. This approach can compensate wrong local decisions: for example, a local `NegEvt` can be compensated by $2\delta + \theta$ indirect `PosEvt` events, and in order to change from `Positive` to `Negative` states (or vice-versa) we need twice the events.

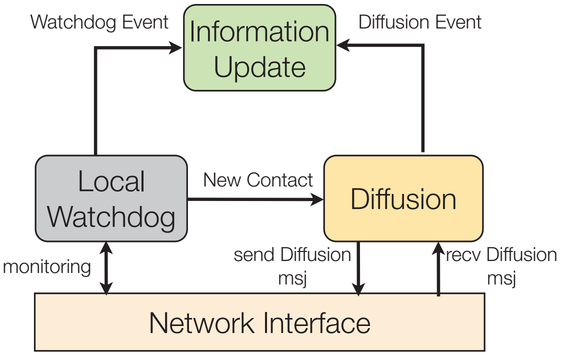


Fig. 2: CoCoWa Architecture

The advantages of this updating strategy are twofold. First, with the threshold θ we can reduce the fast diffusion of false positive and false negatives. Nevertheless, this can produce a delay on the detection (more events are needed to get a *better* decision). Second, the decision about a selfish node is taken using the most recent information. For example, if a node had contact with the selfish node a long time ago (so it had a `Positive` state) and now receives several `NegEvt` in a row from other nodes, the state is updated to `Negative`.

Finally, the network information about the nodes has an expiration time, so after some time without contacts it is updated. The implementation of this mechanism is straightforward. When an event is received, it is marked with a time stamp, so in a given timeout an opposite event is generated, in order to update the value of ρ.

III. Characterising Inter-contact Times

Characterising inter-contact times (or inter-meeting times) between pairs of nodes is essential for analysing the performance of contact-based protocols in cooperative networking. The inter-contact times distribution is obtained by *aggregating* the *individual pair distribution* of all combinations of pairs of nodes in the network. The *individual pair distribution* is defined as the distribution of the time elapsed between two consecutive contacts between the same pair of nodes [27].

The assumption that the aggregated inter-contact time follows an exponential distribution with rate λ has been shown to hold in several mobility scenarios of both humans and vehicles [11], [23], [37]. For example, in [11] it is shown that, for the random waypoint and random direction mobility models, parameter λ is related to the mean speed of nodes v, through an empirical expression. There is some controversy about whether this exponential distribution relates to real mobility patterns. Empirical results have shown that the aggregated inter-contact time distribution follows a power-law and has a long tail [7], meaning that there are some pairs of nodes that barely experience contact. In [6] it is shown that in a bounded domain, such as the one selected along this paper, the inter-contact distribution is exponential, but in an unbounded domain the distribution is power-law. The dichotomy of this distribution is described in [18]: a truncated power law with exponential decay appearing in its tail after some cutoff point. A recent paper [27] presents the dependence between the *individual pair distribution* and the *aggregated distribution*. It is stated that, starting from the exponential *individual pair distribution*, the aggregated is distributed according to a Pareto law. It also verifies the dichotomy property of the aggregate distribution analytically. The work in [10] analysed some popular mobility traces and found that over 85% of the *individual pair distributions* fit an exponential distribution.

Therefore, we consider that using an exponential fit is a valid assumption to model inter-contact times. Our analytical model assumes an exponential distributed inter-contact rate between nodes and, therefore, it is suited for modelling the contacts in MANETs and DTNs networks.

IV. System Model

The network is modelled as a set of N wireless mobile nodes, with C collaborative nodes, M malicious nodes and S selfish nodes ($N = C + M + S$). Our goal is to obtain the time and overhead that a set of $D \leq C$ nodes need to detect the selfish nodes in the network. The overhead is the number of information messages transmitted up to the detection time.

Note that the following models evaluate the detection of a single selfish node. The effect of having several selfish nodes in a network is easy to evaluate, and it does not require a specific model. If we assume that selfish nodes are not cooperative, we can analyse the impact of each selfish node on the network independently. In the case of several selfish nodes ($S > 1$) on a network with N nodes, we can assume that there are $C = N - S$ cooperative nodes.

A. The model for the CoCoWa architecture

The goal of this subsection is to model the behaviour of the different modules of our architecture (see figure 2). The *local watchdog* is modelled using three parameters: the probability of detection p_d, the ratio of false positives p_{fp}, and the ratio of false negatives p_{fn}. The first parameter, the probability of detection (p_d), reflects the probability that, when a node contacts another node, the watchdog has enough information to generate a `PosEvt` or `NegEvt` event. This value depends on the effectiveness of the watchdog, the traffic load, and the mobility pattern of nodes. For example, for Opportunistic Networks or DTNs where the contacts are sporadic and have low duration, this value is lower than for MANETs. Furthermore, the watchdog can generate false positives and false negatives. A false positive is when the watchdog generates a positive detection for a node that is not a selfish node. A false negative is generated when a selfish node is marked as a negative detection. In order to measure the performance of a watchdog, these values can be expressed as a ratio or probability: p_{fp} is the ratio (or probability) of false positives generated when a node contacts a non-selfish node, and p_{fn} is the ratio (or probability) of false negatives generated when a node contacts a selfish node. Using the previous parameters we can model the probability of generating local `PosEvt` and `NegEvt` events when a contact occurs:

- `PosEvt` event: the node contacts with the selfish node and the watchdog detects it, with probability $p_d(1-p_{fn})$. Note that a false positive can also be generated with probability $p_d \cdot p_{fp}$.
- `NegEvt` event: the node contacts with a non-selfish node and detect it with probability $p_d(1-p_{fp})$. A false negative can also be generated when it contacts with the selfish node with probability $p_d \cdot p_{fn}$.

The *diffusion module* can generate indirect events when a contact with neighbour nodes occurs. Nevertheless, a contact does not always imply collaboration, so we model this probability of collaboration as p_c. The degree of collaboration is a global parameter, and it is used to reflect that either a message with the information about the selfish node is lost, or that a node temporally does not collaborate (for example, due to a failure or simply because it is switched off). In real networks, full collaboration ($p_c = 1$) is almost impossible. Finally, the probability of generating the indirect events are the following:

- `PosEvt` event: a contact with another node that has a `Positive` state of the selfish node with probability p_c.
- `NegEvt` event: a contact with another node that has a `Negative` state, being the probability $\gamma \cdot p_c$. Note that not all `Negative` states are transmitted, it depends on the diffusion factor γ.

The *information update* module is driven by the previous local and indirect events. These events update the reputation ρ about a node, and are used to finally decide if a node is selfish or not using the threshold θ.

B. Malicious nodes and attacker model

Malicious nodes attemp to attack the CoCoWa system by generating wrong information about the nodes. Thus, the attacker model addresses the behaviour or capabilities of these malicious nodes. A malicious node attack consists of trying to send a positive about a node that is not a selfish node, or a negative about a selfish node, with the goal of producing false positives and false negatives on the rest of nodes. In order to do this, it must have some knowledge about the way CoCoWa works. The effectiveness of this behaviour clearly depends on the rate and precision that malicious nodes can generate wrong information. Malicious nodes are assumed to have a communications hardware similar to the rest of nodes, so they can hear all neighbour messages in a similar range than the rest of nodes. Nevertheless, the attacker could use high-gain antennas to increase its communications range and thus disseminate false information in a more effective manner.

Regarding the diffusion of information on the network, our approach does not assume any security measures, such as message cyphering or node authentification. Nevertheless, if these measures exist, the effect of malicious nodes in CoCoWo will be very reduced or even non-existent. The diffusion module can also accepts messages from every node, including from malicious ones. Thus, we assume that malicious nodes can be active, and use this information in order to generate wrong positives/negatives about other nodes. Nevertheless, we assume that malicious nodes cannot impersonate other nodes and do not collude with other malicious nodes (that is, they do not cooperate among them). Another problem is the Sybil attack [8]. Since malicious nodes can create and control more than one identity on a single physical device, it can have a serious impact on CoCoWa. Thus, a specific security measure is needed, such as the one presented in [1].

The behaviour of malicious nodes is modeled from the receiver perspective, which is based on the probability of receiving wrong information about a given node when a contact with a malicious node occurs (that is, it receives a `Negative` about the selfish node, and a `Positive` about the other nodes). We denote this behaviour as the *maliciousness probability* p_m. Below we detail several aspects that can affect this probability:

1) The reception of information, considering that not all contacts produce this reception. This aspect is similar to the collaboration degree (that is, the p_c parameter), but an increase of communication range of the malicious nodes will increase the information reception.
2) The malicious nodes do not have information about all nodes; so, in order to send a positive/negative about a node, they must have contacted this node previously or have received a message from other nodes.
3) Another issue to consider is the proper generation of wrong information, for example when receiving a positive of a node that is not a selfish node. From the receiver point of view, a perfect malicious node will always provide wrong information. In this case, the malicious node, in order to send wrong information, must *know* the state of each node. In other words it must have a perfect local watchdog (about the node it contacts).

Summing up, this parameter reflects the average intensity or effectiveness of the attack of the malicious nodes.

C. The model for the detection of selfish nodes

In this subsection we introduce an analytical model for evaluating the performance of CoCoWa. The goal is to obtain the detection time (and overhead) of a selfish node in a network. This model takes into account the effect of false negatives. False positives do not affect the detection time of the selfish node, so p_{fp} is not introduced in this model.

Using λ as the contact rate between nodes, we can model the network using a 4D Continuous Time Markov chain (4D-CTMC). For modelling purposes, the collaborative nodes are divided into two sets: a set with D *destination* nodes, and a set of $E = C - D$ *intermediate* nodes. The *destination* and *intermediate* nodes have the same behaviour (both are collaborative nodes). The only purpose of this division is to analytically obtain the time and the overhead required for the subset of *destination* nodes to detect the selfish node. Thus, the 4D-CTMC states are: $(d_p(t), d_n(t), e_p(t), e_n(t))$, where $e_p(t)$ represents the number of *intermediate* nodes that have a `Positive` state, $e_n(t)$ the *intermediate* nodes with a `Negative` state, $d_p(t)$ the *destination* nodes with a `Positive` state and $d_n(t)$ the *destination* nodes with a `Negative` state. Note that, in this model, a `Negative` is a false negative. The states must verify the following conditions: $d_p(t) + d_n(t) \leq D$ and $e_p(t) + e_n(t) \leq E$. Our 4D-CTMC model has an initial state $(0, 0, 0, 0)$ (that is, all nodes have no information). The final (absorbing) states are when $d_p(t) = D$. We define υ as the number absorbing states, that are all possible permutations of states $(\{(D, 0, *, *)\})$ that sum E. It is easy to derive that $\upsilon = \mathrm{P^S}(E) = 0.5(E + 1)(E + 2)$. The number of transient states τ is obtained in a similar way: $\tau = (\mathrm{P^S}(D) - 1)\mathrm{P^S}(E)$. This model can be expressed using the following generator matrix $\mathbf{Q}$:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{T} & \mathbf{R} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{2}$$

where $\mathbf{T}$ is a $\tau \times \tau$ matrix with elements q_{ij} denoting the transition rate from transient state s_i to transient state s_j, $\mathbf{R}$ is a $\tau \times \upsilon$ matrix with elements q_{ij} denoting the transition rate from transient state s_i to the absorbing state s_j, the left $\mathbf{0}$ is a $\upsilon \times \tau$ zero matrix, and the right $\mathbf{0}$ is a $\upsilon \times \upsilon$ zero matrix.

Now, we derive the transition rates q_{ij}. Given the state $s_i = (e_p, e_n, d_p, d_n)$[1], we have:

$$q_{ij} = \begin{cases} R_p(E - e_p - e_n) & e_p+ \\ R_{fn}(E - e_p - e_n) & e_n+ \\ R_{fn}e_p & e_p- \\ R_p e_n & e_n- \\ R_p(D - d_p - d_n) & d_p+ \\ R_{fn}(D - d_p - d_n) & d_n+ \\ R_{fn}d_p & d_p- \\ R_p d_n & d_n- \end{cases} \tag{3}$$

where $x+$ represents a transition from state $(\cdots, x, \cdots)$ to $(\cdots, x + 1, \cdots)$, and $x-$ represents a transition from state $(\cdots, x + 1, \cdots)$ to $(\cdots, x, \cdots)$. Finally, $q_{ii} = -\sum_{i \neq j} q_{ij}$.

The first transition e_p+ is when a *intermediate* collaborative node changes from `NoInfo` state to a `Positive` state ((d_p, d_n, e_p, e_n) to $(d_p, d_n, e_p + 1, e_n)$). The rate of change depends on the updating of ρ, and on the δ and θ parameters. The reputation value ρ increments according to expression 1. This update can be generated by local events and indirect events. First, the local watchdog can generate a local `PosEvt` with rate $\lambda p_d(1 - p_{fn})$ so the reputation is incremented by δ. Then, the rate of increment due to local events is $\lambda\delta p_d(1 - p_{fn})$. Second, updating from an indirect event depends on the number of nodes with `Positive` and `Negative` states and the probability of collaboration: $\lambda p_c(c_p - \gamma c_n)$ where $c_p = e_p + d_p$ and $c_n = e_n + d_n$. Malicious nodes affect this updating by generating indirect `NegEvt` with a rate $\lambda M p_m$. Since we are evaluating the increment, this term must be positive. So, the final rate due to indirect events is $\lambda \max(p_c(c_p - \gamma c_n) - M p_m)$. All the previous terms are divided by threshold θ in order to obtain the rate of changing when a node contacts with a collaborative node:

$$R_p = \lambda(\delta p_d(1 - p_{fn}) + \max(p_c(c_p - \gamma c_n) - M p_m, 0))/\theta \tag{4}$$

Finally, there are $(E - e_p - e_n)$ nodes with the `NoInfo` state so the final transition rate is $R_p(E - e_p - e_n)$.

[1] For simplicity, we omit the time in the states.

The second transition, e_n+, is when a *intermediate* collaborative node changes from (d_p, d_n, e_p, e_n) to (d_p, d_n, e_p, e_n+1). This means that a *intermediate* collaborative node changes to a `Negative` state (a *false negative*). We can derive a similar expression for the rate of change to a (false) `Negative` state R_{fN}. In this case, when a node contacts with the selfish node, the reputation is decreased with rate $\lambda\delta p_d p_{fn}$, and also by indirect events with rate $\lambda(p_c(\gamma c_n - c_p) + Mp_m)$. Finally, we have:

$$R_{fn} = \lambda(\delta p_d p_{fn} + \max(p_c(\gamma c_n - c_p) + Mp_m, 0))/\theta \quad (5)$$

and the transition is $R_{fn}(E - e_p - e_n)$.

The transition e_p- is when a *intermediate* collaborative node that has a `Positive` state changes to `NoInfo`. This event is similar to e_n+ and the transition rate is similar: $R_{fn}e_p$. Note that in this case we multiply by the number of nodes that have a `Positive` state instead of the number of pending nodes. In a similar way, the transition e_n+ occurs when a *intermediate* collaborative node that has a `Negative` state changes to `NoInfo`. So, the transition rate is $R_p e_n$. For transitions regarding *destination* nodes, the rates are very similar to the previous ones, as seen in expression 3. Finally, all these transitions retain the exponential distribution of *useful contacts* (that is, the contacts that produce a transition), preserving the Markovian nature of the process.

Using the generator matrix $\mathbf{Q}$ we can derive two different expressions: one for the detection time T_d and another for the overall overhead (or cost) O_d. Starting with the detection time, from the 4D-CTMC we can obtain how long it will take for the process to be absorbed. Using the fundamental matrix $\mathbf{N} = -\mathbf{T}^{-1}$, we can obtain a vector $\mathbf{t}$ of the expected time to absorption as $\mathbf{t} = \mathbf{N}\mathbf{v}$, where $\mathbf{v}$ is a column vector of ones ($\mathbf{v} = [1, 1, \ldots, 1]^T$). Each entry t_i of $\mathbf{t}$ represents the expected time to absorption from state s_i. Since we only need the expected time from state $s_1 = (0, 0, 0, 0)$ to absorption (that is, the expected time for all destination nodes to have a `Positive` state), the detection time T_d, is:

$$T_d = E[T] = \mathbf{v_1}\mathbf{N}\mathbf{v} \quad (6)$$

where T is a random variable denoting the detection time for all nodes and $\mathbf{v_1} = [1, 0, \ldots, 0]$. Concerning the overhead we need to obtain the number of transmitted messages for each state s_i. First, the duration of each state s_i can be obtained using the fundamental matrix $\mathbf{N}$. By definition, the elements of the first row of $\mathbf{N}$ are the expected times in each state starting from state 0. Then, the duration of state s_i is $f_i = \mathbf{N}(1, i)$.

Now, we calculate the expected number of messages m_i. The number of messages depends on the diffusion model. For an easier exposition, we start with $\gamma = 0$, that is, only the positive detections are transmitted. From state $s_1 = (0, 0, 0, 0)$ to s_{E+1}=$(0, 0, 0, E)$ no node has a `Positive` state, so no messages are transmitted and $m_1 = 0$. From states $s_{E+2} = (0, 0, 1, 0)$ to $s_{2E+1} = (0, 0, 1, E-1)$, one node has a `Positive` state. In these cases, the `Positive` can be transmitted to all nodes (except itself) for the duration of each state i ($\mathbf{N}(1, i)$) with a rate λ and probability p_c. Then, the expected number of messages can be obtained as $m_i = \mathbf{N}(1, i)\lambda(C-1)p_c$. From states $s_{2E+2} = (0, 0, 2, 0)$ to $s_{3E+1} = (0, 0, 2, E-2)$, we have two possible senders and $m_i = 2\mathbf{N}(1, i)\lambda(C-1)p_c$. Considering both types of nodes (*destination* and *intermediate*), the number of nodes with a `Positive` for state s_i is $\Phi(s_i) = d_p + e_p$. Summarizing, the overhead of transmission (number of messages) is:

$$O_d = E[\text{Msg}] = \lambda(C-1)p_c \sum_{i=1}^{\tau} \Phi(s_i)\mathbf{N}(1, i) \quad (7)$$

Finally, for $\gamma > 0$, the ratio of nodes c_n that will transmit a `Negative` is precisely γ, so $\Phi(s_i) = d_p + e_p + \gamma(d_n + e_n)$.

Using the previous model, we can also evaluate the time when destination nodes D have a "false negative" about the selfish node. In this case the absorbing states are $\{0, D, *, *\}$, that is, when $d_n = D$. A high rate of false negatives and malicious nodes may cause a false negative state to be reached in less time than a true positive detection. This situation (and the solution) is studied in subsection V-B.

D. The model for false positives

We now develop a model for evaluating the effect of false positives. This model evaluates how fast a false positive spreads in the network (the diffusion time). Thus, in this case, a greater diffusion time stands for a lower impact of false positives. The diffusion time is similar to the detection time of true positives described in the previous subsection, and it can be obtained in a similar way. Following the same process that in the previous model for the false negatives, we have a 4D-CMTC with the same states (d_p, d_n, e_p, e_n), but in this case $c_p = d_p + e_p$ represents the number of nodes with a false positive, and $c_n = d_n + e_n$ the number of nodes with a (true) negative detection. We can derive expressions similar to 4 and 5, for the case of false positives. In this case, R_{fP} represents the rate of a false positive, and it is derived in a similar way:

$$R_{fp} = \lambda(\delta p_d p_{fp} + \max(p_c(c_p - \gamma c_n) + Mp_m, 0))/\theta \quad (8)$$

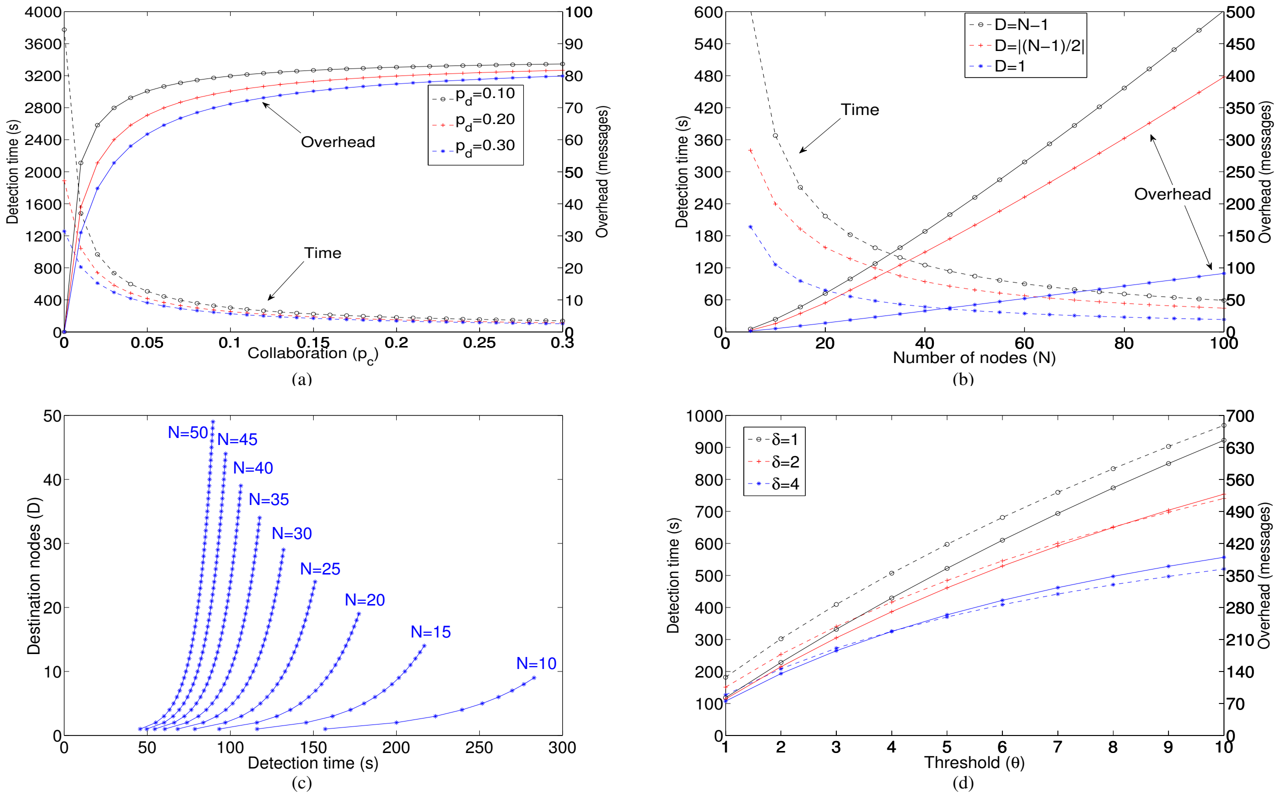
and R_n represents the rate of negative detection:

$$R_n = \lambda(\delta p_d(1 - p_{fp}) + \max(p_c(\gamma c_n - c_p) - Mp_m, 0))/\theta \quad (9)$$

Using these expressions, the transition rates (q_{ij}) of the generator matrix $\mathbf{Q}$ are similar to expression 3, substituting R_P and R_{fn} by R_{fp} and R_n, respectively. Finally, using equations 6 and 7 described in our previous model, we can obtain the diffusion time and the overhead.

V. Analytical Evaluation

This section is devoted to evaluate the performance of CoCoWa. The analytical model introduced in section IV has several parameters, so in this paper we focus on those parameters that clearly affect performance. First, we study the global performance of our approach considering the collaborative issues. Then, we focus our study on the impact of false negatives, false positives, and malicious nodes. Finally, we compare our approach to the classic periodic diffusion model. Note that, since λ is a multiplying factor of all transition rates in matrix $\mathbf{Q}$ (except for q_{ii}), the concluding results of this section are valid for any value of λ (a greater value of λ will affect



(a) (b) (c) (d)

Fig. 3: Global performance evaluation in the absence of false negatives, false positives and malicious nodes. a) detection time depending on collaboration in a network of $N = 25$, b) detection time depending on the number of nodes, c) number of destination nodes that have detected the selfish nodes depending on the detection time, d) detection time depending on the parameters of the detection function (θ and δ). In these plots, the continuous line represents the overhead and the dashed line the detection time.

only on a reduction of the detection time). For the evaluations that follow, we consider a λ value of 0.01 contacts/s, which has been shown to be a valid value in vehicular scenarios [37]. The following evaluations also consider the experimental ranges of several parameters obtained from previous works of our research group [16], [29]. In particular, the probability of detection is low because the local watchdog needs enough packets to generate a positive (or negative) detection of a selfish node $p_d \sim [0.1, 0.3]$, and the ratio of false negatives and false positives are related to p_d; for the range considered the former take the following values: $p_{fn} \sim [0.05, 0.25]$ and $p_{fp} \sim [0.1, 0.3]$.

A. *Global performance evaluation*

In the experiments of this subsection we assume ideal conditions: there are no false positives, no false negatives and no malicious nodes: $p_{fn} = p_{fp} = M = p_m = 0$, and only positive detections are transmitted: $\gamma = 0$. The first evaluation analyse the impact that the degree of collaboration (p_c) has over the efficiency of CoCoWa. The number of selfish nodes is one ($S = 1$) and the detection parameters are: $\theta = \delta = 1$. Figure 3a shows the detection time and overhead for all nodes in a network with 25 nodes ($N = 25, D = 24$) with different probabilities of detection (p_d), ranging from a low detection ratio (0.1), typical of DTNs and Opportunistic Networks, to greater detection ratios (0.3) typical of MANETs [16], [29]. We observe that, when increasing the degree of collaboration from 0 to 0.2, the detection time is reduced exponentially and the overhead is increased. The effect of p_d is the expected: for greater values of p_d, the detection time is reduced. For example, for $p_d = 0.1$, the detection time with no collaboration ($p_c = 0$) is $3775s$. This value can be greatly reduced by using CoCoWa. Thus, even for a low collaboration rate ($p_c = 0.2$), the detection time for all nodes is reduced to $181s$ with an overhead of just 82 messages, which represents an improvement of about 2000% on the detection time. Regarding the detection probability (p_d), we can see that the detection time is greatly reduced even for low values, so CoCoWa is useful in both Opportunistic Networks and DTNs. The previous results show that, when using the local watchdog alone, the detection time is very high (close to one hour). The implications are important. A one hour detection is not useful, because it is equivalent to no detection. Thus, when using collaboration, the detection time is reduced from hours to seconds, meaning that nodes can take appropriate actions in time to avoid the selfish nodes, thereby improving the network performance.

We now evaluate the impact that the number of nodes has on performance. For the following experiments, we set $p_d = p_c = 0.2$. In the first experiment the value of N ranges from 10 to 100 (see figure 3b) while also varying the number of destination nodes. A value of $D = N - 1$ evaluates the detection for all collaborative nodes in the network (the *overall detection*), and $D = 1$ evaluates the detection time for only one node (the *individual detection*). Thus, the *overall detection* evaluates the performance of the entire network, while the *individual detection* evaluates the performance seen from an arbitrary node. We observe that, in general, the greater the number of nodes, the smaller the detection time and the greater the number of messages. The main reason is that, when the number of nodes is greater, the number of contacts increases and so the information about the positive detection is disseminated more quickly. The cost is directly proportional to N.

Finally, as expected, for $D = 1$ the detection time is less than for $D > 1$, but this increase is not exponential with D. We can evaluate this through the dynamics of the overall detection process. Figure 3c shows the number of destination nodes (D) informed about the selfish nodes depending on time with diferent network sizes. The figure shows that, when the number of nodes is low ($N \leq 20$), the first detection takes more time, and the next detections have also a low rate. The reason is that, when N is low, the number of contacts is also low, and so the diffusion of the positives becomes very slow. On the other hand, in a network with more nodes, there are more contacts, meaning that this diffusion is faster (that is, the process runs faster).

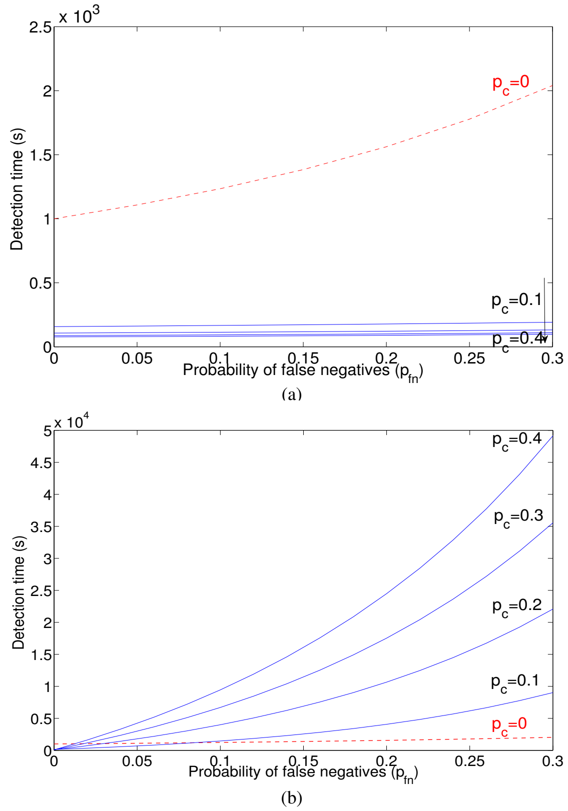
Now, we are going to evaluate the detection function, that is the impact of the θ and δ parameters. We expect that greater θ values imply greater detection times and overhead, due to the number of events required to make a decision. This is confirmed in the results shown in figure 3d. In this plot we can also observe that increasing δ, that is, giving more trust to local events, implies a reduction of both detection time and overhead (which is logical, since less events are needed). The significance of these detection parameters will become more evident when handling malicious nodes.

Finally, the effect of having several selfish nodes $S > 1$ is easy to evaluate. Since the number of cooperative nodes is reduced when S increases ($C = N - S$), the effect is similar to reducing the number of nodes in the network. For example, a network with $N = 100$ and $S = 5$ has a behaviour similar to a network with $N = 96$ and $S = 1$. Thus, the harmful effect of selfish nodes depends mainly on the number of remaining collaborative nodes. If this number is very low (below 20), as shown in figure 3b, the cooperation is greatly reduced and the detection time increases exponentially.

In the following subsections, we evaluate the impact of false negatives, false positives and malicious nodes. Since we are evaluating the performance of the node's collaborative watchdog, we choose to evaluate the performance from an arbitray node (that is, we set $D = 1$).

B. Impact of false negatives

The goal of the following experiments is to evaluate the impact of false negatives. In all the experiments we used $p_d = 0.1$, $M = p_m = 0$, $D = 1$, $N = 25$. We are going to evaluate how the detection time (and overhead) increases depending on the ratio of false negatives (p_{fn}). The first experiment evaluates the influence of collaboration for several values of p_c when only positive detections are transmitted (that is $\gamma = 0$). The detection parameters were $\theta = 1$ and $\delta = 1$. We can see in figure 4a that the detection time increases with the ratio of false negatives. This figure also shows the effect of collaboration: the greater the collaboration the lesser the detection time. This means that, even a low degree of collaboration reduces the impact of these *local* false negatives. Regarding the overhead, the experiment showed little influence on the number of messages, which is always close to 20 messages. Since only positive detections are transmitted, the effect of collaboration is always favourable. Thus, the only effect of increasing the detection parameters (θ) is an increment on the detection time, while it fails at reducing the impact of false negatives.



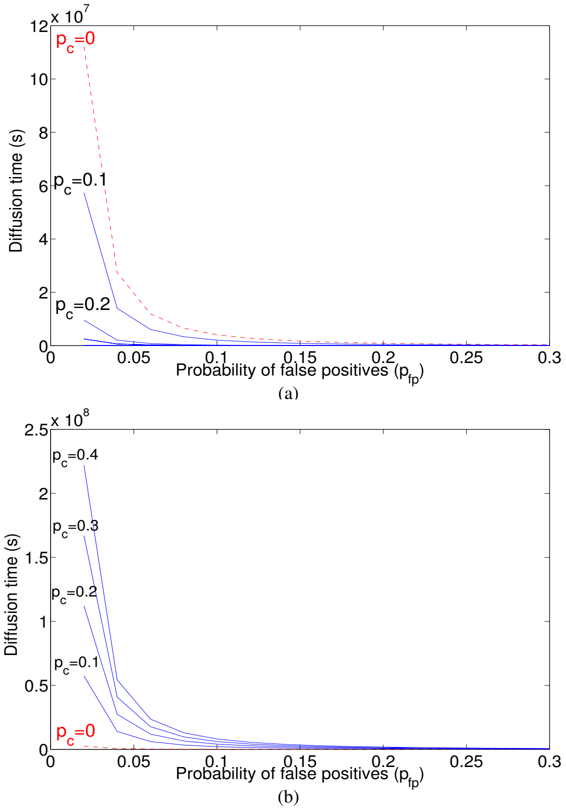
(a)

(b)

Fig. 4: Evaluation of the impact of false negatives, a) for $\gamma = 0$, b) for $\gamma = 1$

Now, we are going to evaluate the effect of transmitting all negative detections ($\gamma = 1$). Figure 4b shows the results for $\gamma = 1$. The results when p_{fn} is zero are very similar to the ”positive detections only” diffusion case ($\gamma = 0$). However, when p_{fn} is greater than zero we can observe that the detection time for values of $p_c > 0$ increases exponentially being greater than the detection time with no collaboration (the dashed red line). We evaluate the time the destination node reaches a false negative state to confirm this effect. When p_{fn} is near to 0.5, the model shows that this false negative state is reached before a true positive state.

Summing up, if only positive detections are transmitted, the detection time is greatly reduced and the impact of false negatives is also reduced; however, when all known negative



(a)

(b)

Fig. 5: Evaluation of the impact of false positives: diffusion time of false positives (the higher the best). a) when only positives are transmitted ($\gamma = 0$), b) when positives and negative are transmitted $\gamma = 1$.

detections are transmitted, collaboration amplifies the effect of false negatives, which is clearly undesirable.

C. Impact of false positives

In this subsection we evaluate the influence of false positives using the model developed in section IV-D. This model evaluates how fast a false positive spreads in the network. Thus, higher values of time imply slower diffusion of false positives. In this case, we expect that the diffusion of negative detections (that is, $\gamma = 1$) will reduce the influence of false positives and that when γ is zero, the influence of false positives will be amplified. Figure 5a shows the diffusion time for $\gamma = 0$, using the same parameters of figure 4a. We observe, that for the curves where $p_c > 0$, the effect of false positives is indeed amplified, leading to a drastic reduction of the diffusion time. This means that these false positives are spread on the network rather quickly, as if they were "true" positives. Consequently, we need to transmit the negative detections in order to compensate for these false positives. Figure 5b shows the results for $\gamma = 1$. In this case, we can see that the detection time is highly increased when the collaboration increases and so the effect of false positives is reduced.

One way to reduce this effect is to increase the reputation threshold θ. The results confirm that the diffusion time is increased and so the harmful impact of false positives is reduced. Nevertheless, the best approach to reduce this effect is to use the diffusion factor. As shown, we have the inverse effect that in the false negatives case. If only positive detections are transmitted the effect of false positives is magnified and so the transmission of negative detections is needed in order to reduce the impact of false positives. This effect can be regulated using the γ factor. Thus, we evaluated the same scenario of figures 4 and 5 for $\gamma = 0.1$. For the detection time the resulting graph is very similar to figure 4a, confirming that the detection time is reduced, even if the ratio of false negatives is high. Regarding the diffusion time, the resulting graph is similar to figure 5a, that is, the diffusion time is increased when the collaboration increases, effectively reducing the effect of false positives. Summing up, the γ value must be tuned properly in order to achieve the desired behavior. A γ value near zero greatly reduces the detection time of selfish nodes, but it increases the diffusion of false positives. A value near one increases the detection time (due to the effect of false negatives), but it reduces the diffusion of false positives. For practical implementations, and based on the results of our experiments, γ values from 0.05 to 0.25 represent good options.

D. Impact of malicious nodes

In the following experiments we evaluate the effect of malicious nodes. Figure 6a shows the detection time of a selfish node depending on the maliciousness probability of one node ($M = 1$). This ratio range from 0 (no malicious behaviour) to 0.5 (a very malicious behaviour). The parameters used are similar to previous experiments ($N = 25$, $D = 1$, $p_{fn} = p_d = 0.1$, $\theta = \delta = 1$, $\gamma = 0.1$). We can conclude that when p_m increases the detection time increases. This effect is reduced for greater degrees of collaboration. Nevertheless, for values of $p_m < 0.3$, the impact is very reduced, meaning that collaboration reduces the impact of malicious nodes. The impact on the diffusion of a false positive is shown in figure 6b when p_{fp} is 0.2. We can see that the diffusion time is reduced when p_m increases, so a false positive has a faster diffusion. Increasing the degree of collaboration reduces this diffusion for low values of p_m.

The previous experiments show that collaboration cannot reduce the impact of malicious nodes for $p_m > 0.2$. Therefore, in order to reduce this impact we need to adjust the values of the detection parameters. In this case, we need to give more trust to the local watchdog (that is, the δ parameter). This is confirmed by the results shown in figure 7a using $p_c = 0.2$. The best results are obtained for $\delta = \theta = 2$ and $\delta = \theta = 3$. Although the detection time is greater compared to $\delta = \theta = 1$ for low values of p_m, when p_m is high, the detection time does not increase exponentially as for $\delta = 1$. Greater values of δ and θ (not shown in the graph), increases the detection time. Thus, given too much trust to the local watchdog is a way to elude collaboration, so the detection time is increased. Finally, regarding the diffusion of false positives, we can see in figure 7b that by increasing θ this diffusion is only slightly reduced.

Finally, the effect of the number of malicious nodes M depends on the number of nodes evaluated. If the ratio M/N is low, the impact can be controlled using collaboration and reputation mechanisms, but if the ratio M/N is high, the

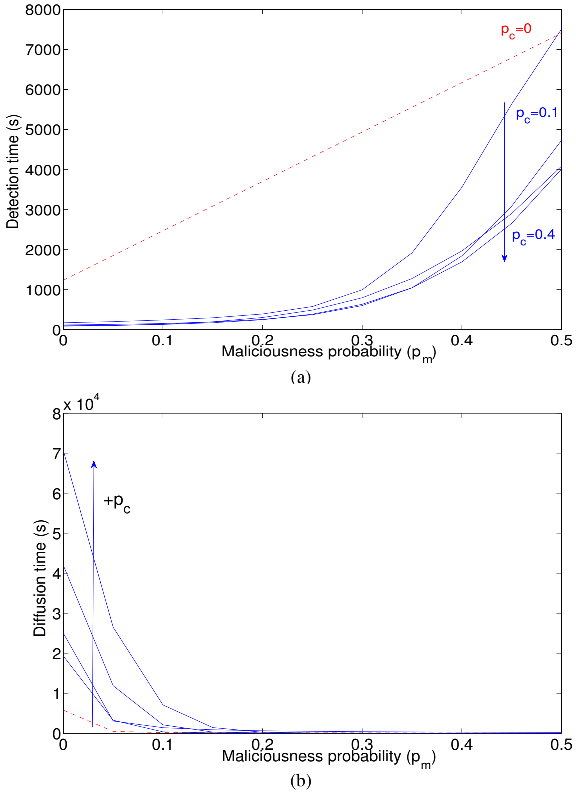


Fig. 6: Impact of malicious nodes a) detection time of selfish node for $\gamma = 0.1$, b) diffusion time of false positives for $\gamma = 0.1$

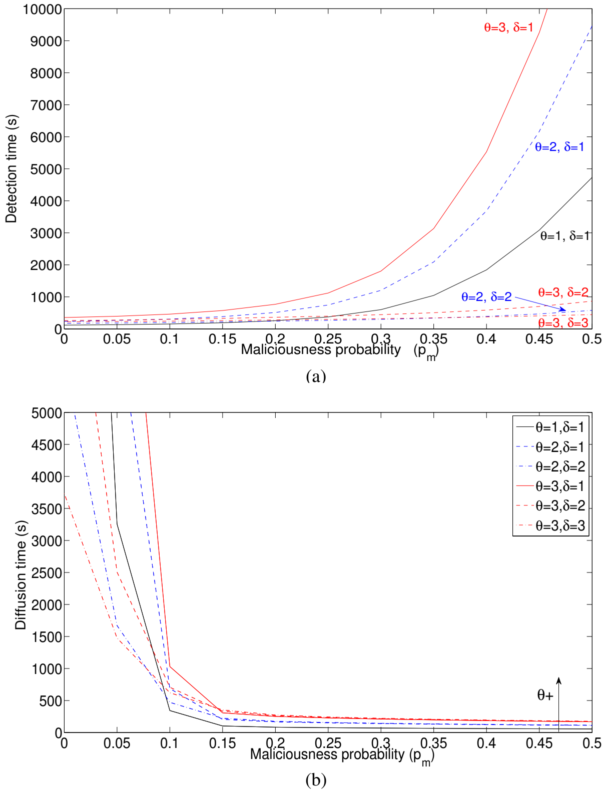


Fig. 7: Reduction of the impact of malicious nodes using different detection parameters a) detection time of selfish nodes, b) diffusion time of false positives.

Performance goal	Parameter tuning
Reduce Detection time	Increase detection ratio (p_d). Increase precision (reduce local False Negative (FN) ratio (p_{fn}))
Reduce Overhead	Indirectly, reducing the detection ratio (p_d). Reduce diffusion factor γ
Reduce Impact of False Positives	Increase precision (reduce local False Positives (FP), p_{fp}). Reduce diffusion factor γ
Reduce Impact of False Negatives	Increase precision (reduce local FN, p_{fn}). Increase diffusion factor γ
Reduce Impact of both FN and FP	Increase precision (reduce local FN and FP). Set diffusion factor γ in [0.05,025] range
Reduce Impact of Malicious Nodes	If the degree of maliciousness is high ($p_m > 0.2$) or/and the number of malicious nodes is high ($M/N > 0.1$), give more trust to local watchdog ($\delta = \theta = \{2, 3\}$)

TABLE I: Criteria for the selection of parameters. This table resumes the main factors that have impact on the consecution of the goals. Some of the factors can be network dependent, so we need to adjust another parameters (if it is possible).

performance of the network can be very low. Our experiments showed that the limit is about 0.1 (that is, one malicious node for each ten collaborative nodes). This contrasts to the effect of selfish nodes, that only depends on the remaining cooperative nodes, and has less impact on network performance. These results are coherent, as they highlight the different behaviour of selfish and malicious nodes.

VI. Experimental Evaluation

This section introduces several experimental results of CoCoWa using two realistic scenarios. It also compares CoCoWa with previous approaches. But first, based on the previous analytical results, we provide some experimental guidelines to optimise CoCoWa.

A. *Guidelines to CoCoWa optimisation*

The goal of this section is twofold: it is a guideline for selecting the correct configuration of CoCoWa for improving the global performance and it also summarises the results obtained in the experiments presented along this paper (and from other experiments not include here).

The main criteria for tuning and adjusting CoCoWa are shown on table I. Each row describes the influence of the different factors on attaining a given performance goal. In general, reducing the impact of false negatives and false positives depends on the γ factor, and in this case, reducing both implies adjusting the diffusion factor, as shown in the table. Regarding the detection parameters, the best results are obtained when $\delta = \theta = 1$. Nevertheless, if the number of malicious nodes or their probability (p_m) is high, we must use $\delta = \theta = \{2, 3\}$ in order to assign more trust to local watchdogs.

Thus, the procedure for CoCoWa optimisation is the following: first, we need to obtain the network characteristics such as number of nodes, contact rate and degree of collaboration. These values can be experimentally measured or estimated. The performance of the local watchdog is also measured (or estimated), and it can depend on the network characteristics. Note that this local watchdog can be adjusted in terms of

	Cambridge	*Shanghai*
Type	Human	Vehicle
Device	iMote	GPS+GPRS
Network Type	Bluetooth	WiFi
Duration (hours)	274	24
Resolution (s)	120	60
Nodes	36	2288
Contacts (C)	21200	1262498
Contact Rate λ (contacts/hour)	0.101	0.012

TABLE II: Description of contact traces. For the inter-contact rates (λ) we used the sames values of [23].

detection and precision (usually, the greater the precision, the lesser the detection ratio, as the local watchdog needs more packet overhearing to generate a more precise detection). For example, in order to reduce the detection time in a network with a given contact rate and collaboration, the only solution is to increase the performance of the local watchdog module (if it is possible). Other network characteristics, such as the number (or ratio) of selfish and malicious nodes, can be evaluated using several scenarios, such as the *worst case* scenario.

B. Real mobility scenarios

In this subsection we are going to evaluate CoCoWa using real mobility scenarios. One of the drawback of the analytical model is the representativity of the mobility model. Although it is shown to be an excellent approximation, it is important to evaluate CoCoWa using both human and vehicular mobility traces. In the following experiments we used some well known real contact traces (see table II). The *Cambridge* mobility set trace [17] was gathered from a set of undergraduate students from the University of Cambridge carrying small devices (iMotes) in 2005. The *Shanghai* Taxis GPS Trace [37] was collected from 2100 taxis in Shanghai city during February of 2007. This trace does not contain the contacts (it contains GPS locations), so a pre-process for obtaining the contact trace is needed. Following the method used in [37] we assume that a contact occurs if both vehicles are within WiFi range (100 meters). The result of processing the previous mobility traces is a contact trace.

We did four experiments with different watchdog and maliciousness parameters for each set of traces (see table III). The parameter N is set to the number of nodes on each network. In all experiments, we obtain the time and overhead for detecting one selfish node ($S = 1$) by one of the nodes in the network ($D = 1$), assuming collaboration ($p_c = 0.3$) and no collaboration ($p_c = 0$), so we can clearly evaluate the benefits of using CoCoWa. The simulator is the one described in appendix A, but in this case we used a real contact trace as the input. For each experiment, we performed 1000 simulations where the destination node and the selfish node were randomly selected from all posible nodes. The final result for each experiment is the mean detection time (and overhead) with confidence intervals. Note that, for the Shanghai experiment, our trace is limited to a 24 hour period. So, in order to simulate more than a day (the mean detection time is greater than 24 hours), we reuse the same trace for every new day, randomly modifying node numbers. This is a way to force all taxis to have a different route every day. Finally, using the contact rate (λ) of each trace (see table II), we also calculated the detection time (and overhead) using the 4D-CTMC analytical model, to check the precision of our model.

Experiment	Parameters
1	No false negatives and positives (low detection ratio), No malicious nodes ($p_{fn} = p_{fp} = 0.0$, $p_d = 0.1$, $M = 0$, $\delta = \theta = 1$, $\gamma = 0.1$)
2	Low ratio of false positives and negatives (higher detection ratio), No Malicious nodes ($p_{fn} = p_{fp} = 0.1$, $p_d = 0.2$, $M = 0$, $\delta = \theta = 1$, $\gamma = 0.1$)
3	Low ratio of false positives and negatives, Low ratio of Malicious nodes ($p_{fn} = p_{fp} = 0.1$, $p_d = 0.2$, $\lceil M = N/50 \rceil$, $p_m = 0.1$, $\delta = \theta = 1$, $\gamma = 0.1$)
4	Worst scenario: Higher ratio of false positives and negatives, Higher ratio of Malicious nodes and maliciousness ($p_{fn} = p_{fp} = 0.2$, $p_d = 0.3$, $\lceil M = N/20 \rceil$, $p_m = 0.1$, $\delta = \theta = 2$, $\gamma = 0.1$)

TABLE III: Parameters of the different real mobility scenarios experiments.

Exp.	Model	Simulation		Simulation (no colab.)	
Cambridge					
1	6.30	7.52	(2.12-12.14)	99.01	(40.36-140.53)
2	4.86	6.15	(2.05-11.02)	61.12	(38.92-99.80)
3	5.51	6.03	(1.12-12.14)	91.68	(43.12-139.60)
4	11.09	7.52	(2.12-12.14)	103.14	(80.48-159.17)
Shanghai					
1	46.82	43.69	(20.06-53.63)	999.39	(614.79-1050.1)
2	36.58	34.93	(18.51-42.11)	605.18	(350.66-775.31)
3	40.77	45.81	(31.26-64.82)	997.77	(592.39-1050.6)
4	93.35	84.91	(50.61-124.5)	1191.45	(634.54-1502.6)

TABLE IV: Detection time in hours using several mobility scenarios. See table III for the parameters of each experiment. In parenthesis are the 95% confidence intervals.

Table IV shows the detection time for the four experiments using both mobility traces. In general, we can see that our approach greatly reduces the detection time of the selfish node compared with a simple local watchdog solution. Even in the worst scenario (low precision watchdog and high ratio of malicious of nodes), the detection is greatly reduced. Regarding the overhead, the results of the previous experiments confirm the analytical results. For example, the overhead in CoCoWa for experiment 2 for Cambridge was 31.5 (12.5-42.2)% using simulation, and 34.2 using the analytical model, which is a very reduced value. In general, the overhead is linear with the number of nodes, so it is a scalable approach.

These experiments confirm the results of the previous section based on the analytical model. We can see that the detection time and the overhead values obtained with the analytical model are close to the simulated ones, so these experiments also validate our analytical model.

C. Comparison with other approaches

We now proceed by comparing the CoCoWa approach with previous cooperative approaches that use periodic messages for the diffusion of information about selfish node detections (such as the ones presented in [20], [26], [28]). Note that this comparison focuses only on the diffusion protocol. If a node has information about a positive (or negative) detection, it will periodically broadcast a message with a given period

P. This message will be received by all nodes that are within the communication range of the sender. The performance of this protocol clearly depends on the period P. A short period will reduce the detection time, but the number of messages transmitted (the overhead) will be high. A large period will increase the detection time by reducing the overhead.

The comparison of both protocols was based on a custom simulator. This simulator reads a mobility trace and, *knowing* the position of the network nodes beforehand, simulates the periodic diffusion protocol, broadcasting a periodic message to all nodes that are within communication range, as described in the previous paragraph. Since our simulator can accept ns-2 *setdest* command mobility traces, we generated different mobility scenarios that are used to simulate both approaches. The main parameters for the mobility model are mean-speed = 5m/s, side-area = 1000 m, pause-interval = 1s and range = 100m. Regarding CoCoWa, the watchdog parameters are ($p_{fp} = 0.17$, $p_{fn} = 0.08$, $p_d = 0.11$), that were obtained based on a set of real testbed experiments from [16]. The remaining parameters are $p_c = 0.2$, $\gamma = 0.1$, $\theta = \delta = 1$ and there are no malicious nodes ($M = 0$).

Figure 8a shows the detection time and overhead for the periodic diffusion protocol when period P ranges from 1 to 30s on a network with 40 nodes. Results confirm that increasing period P implies a higher detection time while reducing the overhead. We compare these results with the detection time and overhead values for CoCoWa. The periodic diffusion for periods below 3s has a shorter detection time than our model, but with a higher overhead. For example, for $P = 1s$, the detection time is 823s (compared with 857s of CoCoWa) and the overhead is 9791 messages (CoCoWa cost is always 162). For $P = 3s$, the detection time is similar to our approach, and the overhead is 3779 messages. In order to clearly compare these approaches, figure 8b shows the ratio between the detection time and overhead for both of them. Three different numbers of nodes ($N = 30, 40, 50$) are used. We can see that, for the periodic diffusion, the detection time increases compared to the CoCoWa approach. Only for reduced periods ($P < 4$) is the detection time lower or equal than for CoCoWa. Regarding the overhead, we can see that even when increasing the period, it is still 6 times greater than with CoCoWa. Regarding false positives, in the periodic model the diffusion time of false positives is reduced for low values of P. For example, for $N = 40$ the detection time of false positives is reduced from $15024s$ when there is no diffusion of positive detections to $900s$ when $P = 1$.

Summarizing, although using periodic diffusion can reduce the detection time slightly, this implies a large overhead and the impact of false positives is very high, and so it is not a viable strategy for low period values.

VII. Related work

There are two main strategies to deal with selfish behaviour in cooperative networks. The first approach tries to motivate the nodes to actively participate in the forwarding activities. For example, in [4], [5] the authors presented a method using a virtual currency called nuglet. Zhong et al. [36] proposed SPRITE, a credit-based system to incentivate participation of selfish nodes in MANET communication. These incentivation methods present several problems, such as the need for some kind of implementation infrastructure to maintain the accounting and they usually rely on the use of some kind of tamper-proof hardware. The COMMIT Protocol [9] combines game-theoretic techniques to achieve truthfulness and an incentivation payment scheme to reduce the impact of selfish nodes on routing protocols. Regarding the detection and exclusion approach, there are several solutions for MANETs and DTNs. A first study about misbehaving nodes and how watchdogs can be used to detect them was introduced in [25]. The authors proposed a Watchdog and Pathrater over the DSR protocol to detect non-forwarding nodes, maintaining a rating for every node. In [28] another scheme for detecting selfish nodes based on context aware information was proposed.

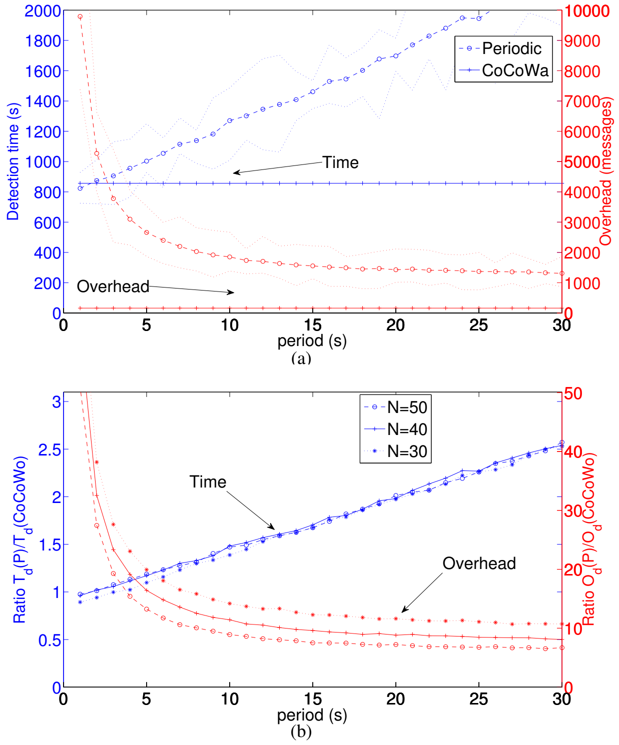


Fig. 8: Comparison of periodic diffusion and CoCoWa. a) Detection time and overhead depending on period P for $N = 40$. The dotted lines around the curves are the 95% confidence intervals. The confidence intervals for the CoCoWa detection time and overhead, are not plotted in the graph and are [625, 1021] and [102, 243] respectively. b) Plot of the ratio between the detection time of periodic diffusion and CoCoWa ($T_d(P)/T_d(CoCoWa)$) and overhead ($O_d(P)/O_d(CoCoWa)$)

In previous works it has been shown how some degree of cooperation can improve the detection of selfish or misbehaving nodes. The CONFIDENT protocol was proposed in [3], which combines a watchdog, reputation systems, bayesian filters and information obtained from a node and its neighbours to securely detect misbehaving nodes. The system's response is to isolate those nodes from the network, punishing then indefinitely. A distributed intrusion detection system (IDS) is introduced in [35]. In this approach if a node locally detects an intrusion with *strong* evidence, it can initiate a response.

However, if a node detects an anomaly with *weak* evidence, it can initiate a cooperative global intrusion detection procedure. A similar approach is the Mobile Intrusion Detection System described in [20]. In this case, local sensor ratings are periodically flooded throughout the network in order to obtain a global rating for each misbehaving node. Another approach is CORE ”Collaborative Reputation Mechanism” [26]. The CORE system is similar to the distributed IDS approaches described below. It consists in local observation using watchdogs that are combined and distributed to obtain a reputation for each node. This reputation is used to determine whether a node is allowed to participate (otherwise, it is excluded). Another approach is OCEAN [2] where the reputation of a neighbour is evaluated using only locally available information, avoiding complex and potentially vulnerable techniques of reputation propagation throughout the network. It is shown that, even with direct neighbour observations, OCEAN performs almost as well as those schemes that share second-hand reputation information. In [14] an analytical selfish model (which is tied specifically to the Ad hoc On-demand Distance Vector (AODV) routing protocol) is proposed. A recent work [34], introduces the Audit-based Misbehaviour Detection (AMD) which isolates continuous and selective packet droppers. The AMD system integrates reputation management, trustworthy route discovery, and identification of misbehaving nodes based on behavioural audits. This scheme also collects first and second-hand information for obtaining the reputation of nodes.

More recently, papers have focused on DTNs. In [19], the author introduces a model for DTN data relaying schemes under the impact of node selfishness. A similar approach is presented in [23] that shows the effect of socially selfish behaviour. Social selfishness is an extension of classical selfishness (also called *individual selfishness*). A social selfish node can cooperate with other nodes of the same group, and it does not cooperate with other nodes outside the group. The impact of social selfishness on routing in DTN has been studied in [22].

Our approach presents similarities with the ones presented in [20], [26]. Nevertheless, these approaches do not evaluate the effect of false positives, false negatives and malicious nodes. For example, the approach in [26] only transmits positive detections. The problem, as shown in the evaluation sections, is that if a false positive is generated it can spread this wrong information very quickly on the network, isolating nodes that are not selfish. Therefore, an approach that includes the diffusion of negative detections as well becomes necessary. Another problem is the impact of colluding or malicious nodes. Although a reputation system, as the one presented in [26], can be useful to mitigate the effect of malicious nodes, it clearly depends on how are combined local and global ratings, as shown in this paper. Another implementation issue is the high imposed overhead due to the flooding process in order to achieve a fast diffusion of the information. Since our approach is based on contacts, it has been proven that the overhead is greatly reduced.

VIII. Conclusions

This paper proposes CoCoWa as a *collaborative contact-based watchdog* to reduce the time and improve the effectiveness of detecting selfish nodes, reducing the harmful effect of false positives, false negatives and malicious nodes. CoCoWa is based on the diffusion of the known positive and negative detections. When a contact occurs between two collaborative nodes, the diffusion module transmits and processes the positive (and negative) detections.

Analytical and experimental results show that CoCoWa can reduce the overall detection time with respect to the original detection time when no collaboration scheme is used, with a reduced overhead (message cost). This reduction is very significant, ranging from 20% for very low degree of collaboration to 99% for higher degrees of collaboration. Regarding the overall precision we show how by selecting a factor for the diffusion of negative detections the harmful impact of both false negatives and false positives is diminished. Finally, using CoCoWa we can reduce the effect of malicious or collusive nodes. If malicious nodes spread false negatives or false positives in the network CoCoWa is able to reduce the effect of these malicious nodes quickly and effectively. Additionally, we have shown that CoCoWa is also effective in Opportunistic Networks and DTNs, where contacts are sporadic and have short durations, and where the effectiveness of using only local watchdogs can be very limited.

In short, the combined effect of collaboration and reputation of our approach can reduce the detection time while increasing the global accuracy using a moderate local precision watchdog.

Acknowledgments

This work was partially supported by the *Ministerio de Ciencia e Innovación*, Spain (grant TIN2011-27543-C03-01).

References

[1] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat. Lightweight sybil attack detection in manets. *Systems Journal, IEEE*, 7(2):236–248, June 2013.

[2] S. Bansal and M. Baker. Observation-based cooperation enforcement in ad hoc networks. arXiv:cs.NI/0307012, 2003.

[3] S. Buchegger and J.-Y. Le Boudee. Self-policing mobile ad hoc networks by reputation systems. *Communications Magazine, IEEE*, 43(7):101 – 107, jul. 2005.

[4] Buttyán, Levente, Hubaux, and Jean-Pierre. Enforcing service availability in mobile ad-hoc WANs. In *Proceedings of MobiHoc'00*, pages 87–96. IEEE Press, 2000.

[5] L. Buttyán and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8:579–592, 2003.

[6] H. Cai and D. Y. Eun. Crossing over the bounded domain: From exponential to power-law intermeeting time in mobile ad hoc networks. *Networking, IEEE/ACM Transactions on*, 17(5):1578 –1591, oct. 2009.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, June 2007.

[8] J. R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[9] S. Eidenbenz, G. Resta, and P. Santi. The COMMIT protocol for truthful and cost-efficient routing in ad hoc networks with selfish nodes. *IEEE Transactions on Mobile Computing*, 7(1):19–33, Jan. 2008.

[10] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *Proceedings of ACM MobiHoc '09*, pages 299–308. ACM, 2009.

[11] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62:210–228, October 2005.

[12] E. Hernández-Orallo, M. D. Serrat, J.-C. Cano, C. M. T. Calafate, and P. Manzoni. Improving selfish node detection in MANETs using a collaborative watchdog. *IEEE Comm. Letters*, 16(5):642–645, 2012.

[13] E. Hernández-Orallo, M. D. Serrat Olmos, J.-C. Cano, C. T. Calafate, and P. Manzoni. Evaluation of collaborative selfish node detection in MANETS and DTNs. In *Proceedings of ACM MSWiM '12*, pages 159–166, New York, NY, USA, 2012.

[14] M. Hollick, J. Schmitt, C. Seipl, and R. Steinmetz. On the effect of node misbehavior in ad hoc networks. In *Proceedings of IEEE International Conference on Communications, ICC'04*, pages 3759–3763. IEEE, 2004.

[15] J. Hortelano, J.-C. Cano, C. T. Calafate, M. de Leoni, P. Manzoni, and M. Mecella. Black hole attacks in p2p mobile networks discovered through bayesian filters. In *Proceedings of P2P CDVE*. Springer, 2010.

[16] J. Hortelano, J. C. Ruiz, and P. Manzoni. Evaluating the uselfusness of watchdogs for intrusion detection in VANETs. In *ICC'10 Workshop on Vehicular Networking and Applications*, 2010.

[17] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of ACM MobiHoc '08*, pages 241–250. ACM, 2008.

[18] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *Proceedings of MobiCom '07*, pages 183–194. ACM, 2007.

[19] M. Karaliopoulos. Assessing the vulnerability of DTN data relaying schemes to node selfishness. *Communications Letters, IEEE*, 13(12):923 –925, december 2009.

[20] F. Kargl, A. Klenk, S. Schlott, and M. Weber. Advanced detection of selfish or malicious nodes in ad hoc networks. In *In Proceedings of Security in Ad-Hoc and Sensor Networks (ESAS 2004*, pages 152–165. Springer Verlag, 2004.

[21] F. Kargl, A. Klenk, M. Weber, and S. Schlott. Sensors for detection of misbehaving nodes in MANETs. In *Proceedings of Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, 2004.

[22] Q. Li, S. Zhu, and G. Cao. Routing in socially selfish delay tolerant networks. In *Proceedings of INFOCOM'10*, pages 857–865. IEEE Press, 2010.

[23] Y. Li, G. Su, D. Wu, D. Jin, L. Su, and L. Zeng. The impact of node selfishness on multicasting in delay tolerant networks. *Vehicular Technology, IEEE Transactions on*, 60(5):2224 –2238, jun 2011.

[24] M. Mahmoud and X. Shen. ESIP: Secure incentive protocol with limited use of public-key cryptography for multihop wireless networks. *IEEE Transactions on Mobile Computing,*, 10(7):997 –1010, july 2011.

[25] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of ACM MobiCom '00*, pages 255–265. ACM, 2000.

[26] P. Michiardi and R. Molva. CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, 2002. Kluwer, B.V.

[27] A. Passarella and M. Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *Proceedings of IFIP NETWORKING'11*, pages 301–313. Springer-Verlag, 2011.

[28] K. Paul and D. Westhoff. Context aware detection of selfish nodes in DSR based ad-hoc networks. In *Proceedings of IEEE Globecom*, 2002.

[29] M. D. Serrat-Olmos, E. Hernández-Orallo, J.-C. Cano, C. T. Calafate, and P. Manzoni. A collaborative bayesian watchdog for detecting black holes in MANETs. In *Intelligent Distributed Computing VI*, volume 446, pages 221–230. Springer, 2012.

[30] C. K. N. Shailender Gupta and C. Singla. Impact of selfish node concentration in MANETs. *International Journal of Wireless and Mobile Networks (IJWMN)*, 3(2):29–37, Apr 2011.

[31] C. Toh, D. Kim, S. Oh, and H. Yoo. The controversy of selfish nodes in ad hoc networks. In *Proceedings of Advanced Communication Technology (ICACT)*, volume 2, pages 1087 –1092, feb. 2010.

[32] Y. Yoo, S. Ahn, and D. Agrawal. A credit-payment scheme for packet forwarding fairness in mobile ad hoc networks. In *Proceedings of IEEE ICC*, volume 5, pages 3005 – 3009 Vol. 5, may 2005.

[33] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867 – 2891, 2007.

[34] Y. Zhang, L. Lazos, and W. Kozma. AMD: Audit-based misbehavior detection in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, PP(99):1, 2012.

[35] Y. Zhang, W. Lee, and Y.-A. Huang. Intrusion detection techniques for mobile wireless networks. *Wirel. Netw.*, 9(5):545–556, Sept. 2003.

[36] S. Zhong, J. Chen, and Y. Yang. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of INFOCOM' 03*, volume 3, pages 1987 – 1997 vol.3, mar. 2003.

[37] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni. Recognizing exponential inter-contact time in VANETs. In *Proceedings of INFOCOM'10*, pages 101–105. IEEE Press, 2010.

Enrique Hernández-Orallo is an associate professor in the Department of Computer Engineering at the Universitat Politècnica de València (UPV) in Spain. He earned an MSc and a Ph.D. in Computer Science from the UPV in 1992 and 2001 respectively. From 1991-2005 he worked at several companies in real-time and computer networks projects. His areas of interest include distributed systems, performance evaluation and mobile and pervasive computing. He is a member of the IEEE.

Manuel David Serrat Olmos received a Degree in Computer Science in 1995 from the UPV, a Master in ITC Department Managing from the Universidad Politécnica de Madrid (UPM) in 2008, a MSc in Computer Engineering from the UPV in 2011 and the PhD in computer science form the UPV in 2013. He worked for several institutions and, nowadays, he is ITC Director at the Valencia County Fire Department. He is co-author of about ten journal and conference papers, and author of one Linux book.

Juan-Carlos Cano is a full professor in the Department of Computer Engineering at the Universitat Politècnica de València (UPV) in Spain. He earned an MSc and a Ph.D. in Computer Science from the UPV in 1994 and 2002 respectively. From 1995-1997 he worked as a programming analyst at IBM's manufacturing division in Valencia. His current research interests include Vehicular Networks, Mobile Ad Hoc Networks, and Pervasive Computing.

Carlos T. Calafate is an associate professor in the Department of Computer Engineering at the Universitat Politècnica de València (UPV) in Spain. He graduated with honors in Electrical and Computer Engineering at the University of Oporto (Portugal) in 2001. He received his Ph.D. degree in Computer Science from the Technical University of Valencia in 2006, where he has worked since 2005. His research interests include mobile and pervasive computing, security and QoS on wireless networks, as well as video coding and streaming.

Pietro Manzoni received the MS degree in computer science from the "Università degli Studi" of Milan, Italy, in 1989, and the PhD degree in computer science from the "Politecnico di Milano", Italy, in 1995. He is currently a full professor of computer science at the Universitat Politècnica de València, Spain. His research activity is related to Mobile Wireless Data Systems design, modelling, and implementation. He is member of the IEEE.

Appendix
Model validation

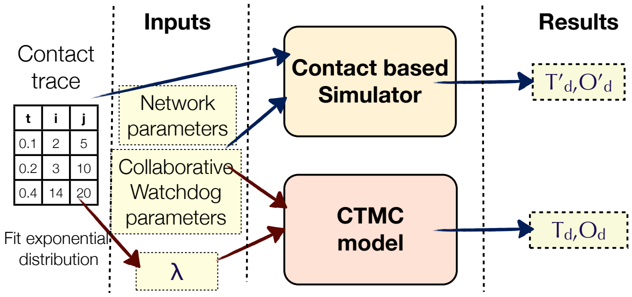
In this appendix we validate the models presented in section IV. The validation procedure is similar to the one described in [33]. We compare the results obtained with our analytical model with those obtained using a simulator. The custom simulator is driven by contacts and uses the same parameters of the network model (N, C, D, S, M) and CoCoWa (p_d, p_c, p_{fp}, p_{fn}, p_m, γ, δ, θ). This simulator reads the contact trace and simulates the behaviour of the watchdog and diffusion modules to change the state of a node. The simulation finishes when all the destination nodes D have a `Positive` state, obtaining the simulated detection time and cost (T'_d, O'_d).

Figure 9 shows our validation process. A contact trace is generated following the same model that was used in [33]. From the contact trace we fit the exponential distribution in order to obtain the λ value required as the input of our analytical model. We set the network and CoCoWa model parameters to obtain the detection time and overhead (T_d,O_d). Using the same network parameters we obtain time and overhead using the contact-based simulation. This simulation is repeated 1000 times in order to obtain a confidence interval for the mean detection time and overhead ($\overline{T'_d}$, $\overline{O'_d}$).

The validation process was based on a set of R repeated random tests. The tests have different parameter values that are randomly generated from a defined range of possible values (see table V). For each test i, a *relative modelling error* of the detection time and cost were obtained ($\epsilon(i)_{T_d}$, $\epsilon(i)_{O_d}$):

$$\epsilon(i)_{T_d} = \frac{T_d(i) - \overline{T'_d(i)}}{T_d(i)}, \qquad \epsilon(i)_{O_d} = \frac{O_d(i) - \overline{O'_d(i)}}{O_d(i)} \quad (10)$$

After running 1000 tests we obtained the mean error (and 95% confidence intervals). For the detection time the mean relative error was 4.32 (0.31-8.74)%, and for the overhead it was 5.92 (0.31-11.15)%. These results validate the model proposed in this paper. Using this simulator, we also validated that the distribution between transitions follows an exponential distribution, preserving the Markovian nature of the process. Finally, our model was also validated using real mobility traces, as shown in subsection VI-B.



t	i	j
0.1	2	5
0.2	3	10
0.4	14	20

Fig. 9: Validation of the model. The input of the simulator and the models are the network parameters and the system parameters.

Parameters	Range
Random Waypoint Model	
Node speed (v)	$\mathcal{U}(5, 15)$
Communication Range (r)	$\mathcal{U}(100, 250)$
Side area (l)	$\mathcal{U}(500, 1500)$
Walk time (w)	$\mathcal{U}(10, 200)$
Network Parameters	
Nodes (N)	$\mathcal{I}(5, 100)$
Selfish nodes (S)	$\mathcal{I}(1, \lceil N/5 \rceil)$
Malicious nodes (M)	$\mathcal{I}(0, \lceil N/5 \rceil)$
Collaborative nodes (C)	$N - M - S$
Destination nodes (D)	$\mathcal{I}(1, C)$
CoCoWa Parameters	
Probability of detection (p_d)	$\mathcal{U}(0.05, 0.3)$
Collaboration degree (p_c)	$\mathcal{U}(0.05, 0.3)$
Rate of false positives (p_{fp})	$\mathcal{U}(0, 0.3)$
Rate of false negatives (p_{fn})	$\mathcal{U}(0, 0.25)$
Maliciousness Probability (p_m)	$\mathcal{U}(0, 0.5)$
Diffusion factor (γ)	$\mathcal{U}(0, 1)$
Local trust factor (δ)	$\mathcal{I}(1, 5)$
Threshold (θ)	$\mathcal{I}(0, 5) + \delta$

TABLE V: Validation scenarios. $\mathcal{U}(a, b)$ stands for the uniform distribution (over interval (a, b)) and $\mathcal{I}(a, b)$ for an uniform integer distribution.