A COLLABORATIVE CONTACT-BASED WATCHDOG FOR DETECTING SELFISH NODES IN COOPERATIVE MANET

Momin Kashif Mukhtar[1]

[1]*III semester, M.E (Computer Network), Computer Department, Sinhgad Institute of Technology, Lonavala, Maharashtra, India*

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, this paper 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.

Keywords: *Opportunistic and Delay Tolerant Networks, Performance Evaluation, Selfish Nodes Wireless networks, MANETs.*

----------***----------

1. 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 impact of node selfishness on MANETs has been studied in [3]. In [2] 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 shows similar results: the number of packet losses is increased by 500% when the selfish node ratio increases from 0% to 40%.

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. 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 Contactbased 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.

2. 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. 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. By using this technique, the local watchdog can generate a positive (or negative) detection in case the node is acting selfishly (or not).

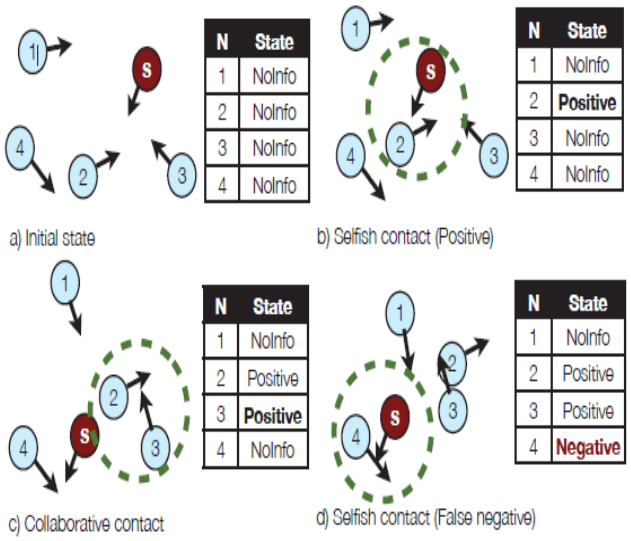

Fig. 1: An example of how CoCoWa works. a) Initially all nodes have no information about the selfish node. b) Node 2 detects the selfish node using its own watchdog. c) Node 2 contacts with node 3 and it transmits the positive about the selfish node. d) The local watchdog of Node 4 fails to detect the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

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

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.

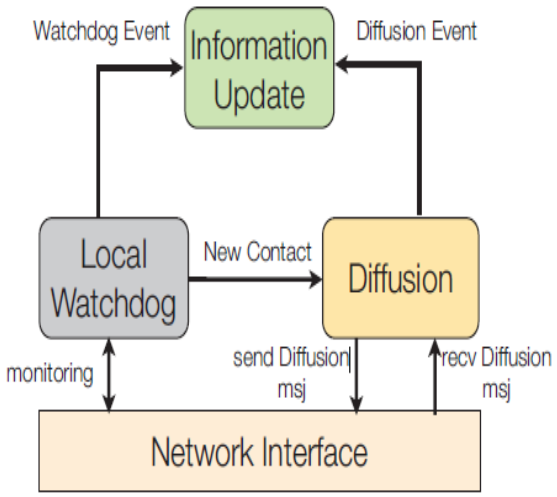

Fig. 2: CoCoWa Architecture

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.

3. 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.

3.1 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 pc. 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 pc.
• 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.

3.2 Malicious Nodes and Attacker Model

Malicious nodes attempt 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 accept 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 [2]. 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 pc 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.

3.3 The Model for the Detection of Selfish Node

In this subsection 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 (4DCTMC). 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 $\upsilon = P^S(E) = 0.5(E + 1)(E + 2)$. The number of transient states τ is obtained in a similar way:

$\tau = (P^S(D) - 1)P^S(E)$. This model can be expressed using the following generator matrix Q:

$$Q = \begin{pmatrix} T & R \\ 0 & 0 \end{pmatrix} \quad (2)$$

where 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 , 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 0 is a $\upsilon \times \tau$ zero matrix, and the right 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:

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

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 λMpm. Since we are evaluating the increment, this term must be positive. So, the final rate due to indirect events is $\lambda\max(p_c(c_p - c_n) - \text{Mpm})$. 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:

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

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

The second transition, en+, 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 RfN. 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 $\lambda(pc(\gamma c_n - c_p) + Mp_m)$. 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 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 $N = -T^{-1}$, we can obtain a vector t of the expected time to absorption as $t = N_v$, where v is a column vector of ones ($v = [1, 1, \ldots, 1]^T$). Each entry t_i of t represents the expected time to absorption from state si. Since we only need the expected time from state s1 = (0, 0, 0, 0) to absorption (that is, the expected time for all destination nodes to have a Positive state), the detection time Td, is:

$$T_d = E[T] = v_1 N_v \quad (6)$$

where T is a random variable denoting the detection time for all nodes and $v_1 = [1, 0, \ldots, 0]$. 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 N. By definition, the elements of the first row of N are the expected times in each state starting from state 0. Then, the duration of state s_i is $f_i = N(1, i)$. 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 m1 = 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 (N(1, i)) with a rate λ and probability p_c. Then, the expected number of messages can be obtained as m_i = N(1, i)λ(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 = 2N(1, i)λ(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:

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

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

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.

3.4 The Model for False Positives

Here introduces 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:

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

and Rn represents the rate of negative detection:

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

Using these expressions, the transition rates (q_{ij}) of the generator matrix 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.

4. CONCLUSION

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. 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.

The combined effect of collaboration and reputation of this approach can reduce the detection time while increasing the global accuracy using a moderate local precision watchdog.

REFERENCES

[1]. Enrique Hern´andez-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni. CoCoWa: A Collaborative Contact-based Watchdog for Detecting Selfish Nodes. IEEE Transactions on Mobile Computing, June 2014.

[2]. 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.

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

[4]. 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.

[5]. E. Hern´andez-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.