International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

CONTACT DISSEMINATION BASED COLLABORATIVE WATCHDOG APPROACH TO IMPROVE SELFISH NODE DETECTION IN MANETS

M. Annie Sharmila[1], Dr.G. Murugaboopathi[2]
[1]PG Student, Network Engineering, Vel Tech Multi Tech Engineering College, Avadi, Chennai.
[2]Associate Professor, Information Technology,Vel Tech Multi Tech Engineering College, Avadi, Chennai.
Email:anniesep2@gmail.com
Email:muruganchitra@gmail.com

Abstract

Mobile ad hoc networks are composed of mobile nodes connected by wireless links without any pre-existing infrastructure. MANET nodes rely on network cooperation schemes to properly work, forwarding traffic unrelated to its own use. However, in the real world, most nodes may have a selfish behaviour, being unwilling to forward packets for others in order to save resources. The selfish nodes are not malicious but are reluctant to spend their resources such as CPU time, memory and battery power for others. Therefore, detecting these nodes is essential for network performance.

Watchdogs are used to detect selfish nodes in computer networks. A way to reduce the detection time and to improve the accuracy of watchdogs is the collaborative approach. This paper proposes a collaborative watchdog based on contact dissemination of the detected selfish nodes. Then, we introduce an analytical model to evaluate the detection time and the cost of this collaborative approach. Numerical results show that our collaborative watchdog can dramatically reduce the overall detection time with a reduced overhead.

***Keywords*-- MANET, Selfish node ,watchdog ,contact dissemination**

I. INTRODUCTION

Mobile ad hoc network is a self configuring infra structureless network of mobile devices connected by wireless links. Ad hoc in latin means " for this purpose ". Each device in a MANET is free to move independently in any direction and will change its links to other devices frequently. Each must forward traffic unrelated to its own use and act as a router. The primary challenge in building a MANET is to equip a device to continuously maintain the information required to properly route the traffic. They may operate themselves.

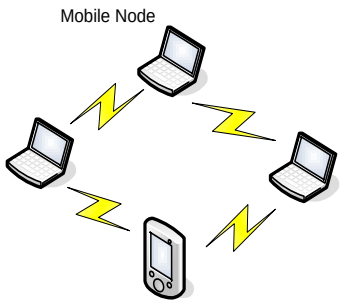

Fig 1.Mobile Ad Hoc Network

MANETs are used in various contexts like intelligent transportation systems, mobile social networks, emergency deployment, etc. In a MANET, nodes can freely move around while communicating with each other. These networks may under-perform in the presence of nodes with a selfish behaviour, particularly when operating under energy constraints. A selfish node will typically not cooperate in the transmission of packets, seriously affecting network performance. Although less frequent, nodes may also fail to cooperate either intentionally (a malicious behaviour) or due to faulty software or hardware.

As there is no dedicated infrastructure or central coordination, the nodes have to cooperate and self-organize to form a working communication network.

Communication only works if nodes participate and forward other node's packets. On the other hand every node has to consider its limited resources (most notably its energy). So every node is motivated to contribute as litte as possible of its own energy. Usually, it is expected that all nodes forward as needed, but other policies are possible as well (e. g. only require forwarding as long as a node's battery level is high). In any way the MANET's protocols and policies imply a normative expectation on every participating node a) to behave according to agreed protocols and b) to forward a fair amount of other node's packets as needed.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

As long as all nodes adhere to this and cooperate, the MANET should work without problems. One of the most important issues in designing MANET protocols is how to deal with nodes that do not cooperate.

II. Wireless Networks

A wireless network in general consists of a set of mobile hosts which communicate to other mobile hosts either directly or via an access point (base station). Wireless network refers to any type of computer network that is not connected by cables of any kind. It is a method by which homes, telecommunications networks and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Wireless telecommunications networks are generally implemented and administered using radio communication. This implementation takes place at the physical level (layer) of the OSI model network structure.

A Wireless Local Area Network (WLAN) consists of a set of mobile users communicating via a fixed base station or an access point. The mobile node can be any device such as a palmtop, PDA, laptop etc. Such networks are usually deployed in offices, cafeterias, universities, etc. and are most prevalently used nowadays. There are three types of WLANs – Independent Basic Service Set (IBSS), Basic Service Set (BSS) and Extended Service Set (ESS). A detailed classification is beyond the scope of this thesis. IEEE 802.11 is an adopted international standard for wireless LANs which provides transmission speeds ranging from 1 Mbps to 54 Mbps in either the 2.4 GHz or 5 GHz frequency bands. The latest version of this standard in use today is IEEE 802.11g which provides a bandwidth of up to 54 Mbps.

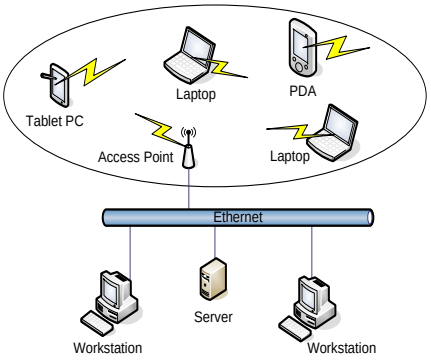

Fig 2 Wireless LAN

III. Misbehaviour In Ad Hoc Networks

Ad hoc networks have a wide variety of commercial applications. Ad hoc networks are ideal in situations where installing an infrastructure is not possible because the infrastructure is too expensive or too vulnerable, the network is too transient, or the infrastructure was destroyed. For example, nodes may be spread over too large an area for one base station and a second base station may be too expensive.

An example of a vulnerable infrastructure is a military base station on a battlefield. Networks for wilderness expeditions and conferences may be transient if they exist for only a short period of time before dispersing or moving. Finally, if network infrastructure has been destroyed due to a disaster, an ad hoc wireless network could be used to coordinate relief efforts. Since DARPA's PRNET, the area of routing in ad hoc networks has been an open research topic.

Ad hoc networks maximize total network throughput by using all available nodes for routing and forwarding. Therefore, the more nodes that participate in packet routing, the greater the aggregate bandwidth, the shorter the possible routing paths, and the smaller the possibility of a network partition. However, a node may misbehave by agreeing to forward packets and then failing to do so, because it is overloaded, selfish, malicious, or broken.

An overloaded node lacks the CPU cycles, buffer space or available network bandwidth to forward packets. A selfish node is unwilling to spend battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to it, even though it expects others to forward packets on its behalf. A malicious node launches a denial of service attack by dropping packets. A broken node might have a software fault that prevents it from forwarding packets.

Misbehaving nodes can be a significant problem. Simulations show that if 10%-40% of the nodes in the network misbehave, then the average throughput degrades by 16%- 32%. However, the worst case throughput experienced by any one node may be worse than the average, because nodes that try to route through a misbehaving node experience high loss while other nodes experience no loss. Thus, even a few misbehaving nodes can have a severe impact.

IV. Selfish Nodes

Selfishness can be termed as a node that doesn't perform its duty. Selfishness in our context can be expressed in two ways. Firstly nodes may deny copying and storing data, which are of no interest to them and destined to a third node.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Secondly even if they accept to acquire such data, they may refuse to infect another node with them, i.e. relay data to other nodes Node selfishness means, that the nodes do not copy or store the data which is intended for a third node. If at all they copy the data they don't relay the data to other node. Varying degrees of selfishness is also exhibited by these nodes. So the performance of these two routing protocols is assessed analytically using a Continuous Time Markov Chain (CTMC). The selfish nodes are modelled as an absorbing two dimensional Continuous Time Markov Chain from which the expected delay is derived. The same model is used to find the delay when there are no selfish nodes.

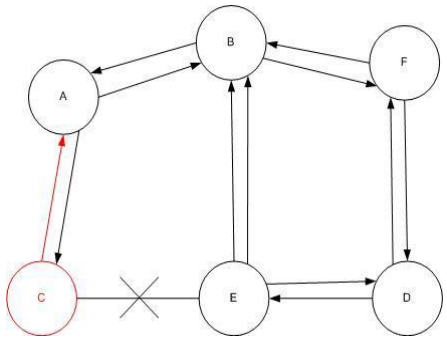

Fig 3 C is the selfish node

V. DATA TRANSMISSION

Malicious node detects the active route and notes the destination address. Malicious node sends a route reply packet (RREP) including the destination address field spoofed to an unknown destination address. Hop count value is set to lowest values and the sequence number is set to the highest value. Malicious node send RREP to the nearest available node which belongs to the active route. This can also be send directly to the data source node if route is available. The RREP received by the nearest available node to the malicious node will relayed via the established inverse route to the data of source node. The new information received in the route reply will allow the source node to update its routing table. New route selected by source node for selecting data. The malicious node will drop now all the data to which it belong in the route.

VI. BAYESIAN WATCHDOG

The watchdog method detects misbehaving nodes. Suppose there exists a path from node S to D through intermediate nodes A, B, and C. Node A cannot transmit all the way to node C, but it can listen in on node B's traffic. Thus, when A transmits a packet for B to forward to C, A can often tell if B transmits the packet.

If encryption is not performed separately for each link, which can be expensive, then A can also tell if B has tampered with the payload or the header.

We implement the watchdog by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet.

If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. The watchdog technique has advantages and weaknesses. DSR with the watchdog has the advantage that it can detect misbehavior at the forwarding level and not just the link level. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping.

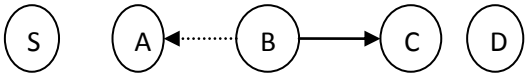

Fig 4 When B forwards a packet from S toward D through C, A can overhear B's transmission and can verify that B has attempted to pass the packet to C. The solid line represents the intended direction of the packet sent by B to C, while the dashed line indicates that A is within transmission range of B and can overhear the packet transfer.

Another problem is that a misbehaving node that can control its transmission power can circumvent the watchdog. A node could limit its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the true recipient. This would require that the misbehaving node know the transmission power required to reach each of its neighbouring nodes. Only a node with malicious intent would behave in this manner selfish nodes have nothing to gain since battery power is wasted and overloaded nodes would not relieve any congestion by doing this.

Finally, a node can circumvent the watchdog by dropping packets at a lower rate than the watchdog's configured minimum misbehavior threshold. Although the watchdog will not detect this node as misbehaving, this node is forced to forward at the threshold bandwidth.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

In this way the watchdog serves to enforce this minimum bandwidth.

The watchdog mechanism could be used to some degree to detect replay attacks but would require maintaining a great deal of state information at each node as it monitors its neighbors to ensure that they do not retransmit a packet that they have already forwarded. Also, if a collision has taken place at the receiving node, it would be neccesary and correct for a node to retransmit a packet, which may appear as a replay attack to the node acting as its watchdog.

Therefore, detecting replay attacks would neither be an efficient nor an effective use of the watchdog mechanism. For the watchdog to work properly, it must know where a packet should be in two hops. In our implementation, the watchdog has this information because DSR is a source routing protocol. If the watchdog does not have this information (for instance if it were implemented on top of a hop-by-hop routing protocol), then a malicious or broken node could broadcast the packet to a non-existant node and the watchdog would have no way of knowing. Because of this limitation, the watchdog works best on top of a source routing protocol.

VII. COLLABORATIVE WATCHDOG

This paper introduces an efficient approach to reduce the detection time of selfish nodes based on contact dissemination. If one node has previously detected a selfish node using its watchdog it can spread this information to other nodes when a contact occurs. We say that a node has a positive if it knows the selfish node. The detection of contacts between nodes is straightforward using the node's watchdog. Notice that the watchdog is overhearing the packets of the neighbourhood; thus, when it starts receiving packets from a new node it is assumed to be a new contact. Then, the node transmits one message including all known positives it knows to this new contacted node. The number of messages needed for this task is the overhead of the collaborative watchdog.

Formally, we have a network of N wireless mobile nodes, with C collaborative nodes and S selfish nodes. Initially, the collaborative nodes have no information about the selfish nodes. A collaborative node can have a positive when a contact occurs in the following way:

7.1. Selfish Contact :

One of the nodes is the selfish node. Then the collaborative node can detect it using its watchdog and have a positive about this selfish node. Nevertheless, a contact does not always imply detection. To model this fact, we introduce a probability of detection (p_d).

This probability depends on the effectiveness of the watchdog and the type of contact (for example if the contact time is very low, the watchdog does not have enough information to evaluate if the node is selfish or not).

7.2. Collaborative Contact :

Both nodes are collaborative. Then, if one of them has one or more positives, it can transmit this information to the other node; so, from that moment, both nodes have these positives. As in the selfish contact case, a contact does not always imply collaboration. We model this with the probability of collaboration (p_c). The degree of collaboration is a global parameter of the network to be evaluated. This value is used to reflect that either a message with the information about the selfish nodes 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.

Although defining a reaction scheme is out of the scope of this paper, there are basically two approaches in the literature: isolation and incentivation. Isolation methods are intended to keep the misbehaving nodes outside the network, excluding them from all kinds of communication. Incentivation methods try to convince the selfish nodes to change their behaviour, and become collaborative instead of selfish, using a virtual payment scheme or a similar mechanism.

VIII. PERFORMANCE MODEL

The goal of this section is to obtain a model to evaluate the time and cost of detecting selfish nodes on a network with collaborative watchdogs. The network is modelled as a set of N wireless mobile nodes, with C collaborative nodes and S selfish nodes ($N = C + S$). It is assumed that the occurrence of contacts between two nodes follows a Poisson distribution λ. This assumption has been shown to hold in several mobility scenarios of both human and vehicles [9]-[12]. For example, in [9] a useful expression is derived for obtaining λ from the parameters of the random waypoint and random direction models.

First, we derive a basic model for $S = 1$. In this case, a collaborative node has 2 states: NOINFO, when the node has no information about the selfish node, and POSITIVE when the node knows who the selfish node is (it has a positive). All nodes have an initial state of NOINFO and they can change their initial state when a contact occurs. Using a contact rate λ we can model the network using a Continuous Time Markov Chain (CTMC) with states $s_i = (c)$, where c represents the number of collaborative nodes in the POSITIVE state.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

At the beginning, all nodes are in NOINFO state. Then, when a contact occurs, c can increase by one. The final (absorbing) state is when c = C. So, this can be modelled using a CTMC with an initial state $s_1 = (0)$, τ = C transient states, and one ($\upsilon = 1$) absorbing state $s_{\tau+1}$ = (C + 1). Then, the transition matrix P in canonical form is:

$$P = \begin{matrix} Q & R \\ 0 & I \end{matrix}$$

Where I is a $\upsilon \times \upsilon$ identity matrix (in this case 1), 0 is a $\upsilon \times \tau$ zero matrix, Q is a $\tau \times \tau$ matrix with elements p_{ij} denoting the transition rate from transient state s_i to transient state s_j and R is a $\tau \times \upsilon$ matrix with elements p_{ij} denoting the transition rate from transient state s_i to the absorbing state s_j.Now, we derive the transition rates p_{ij} . Given a state $s_i = (c)$ the following transitions can occur:

• (c) to (c + 1): This case takes place when a collaborative node changes from NOINFO state to POSITIVE state.The transition probability is $t_c = (\lambda p_d + \lambda p_c c)(C - c)$. The term λp_d represents the probability of detection of a selfish node (using the watchdog) and $\lambda p_c c$ the probability of transmission for the information of the selfish node (it depends on c, so this probability is greater if more nodes are in the POSITIVE state). Finally, factor (C − c) represents the number of pending nodes.

• (c) to (c): This is the probability of no changes, and its value is $t_0 = 1 - t_c$.Using the transition matrix P we can derive two different expressions: one for the detection time T_d and another one for the overall overhead (or cost) M_d. We start with the detection time. Using the fundamental matrix $N = (I - Q)^{-1}$, we can obtain a vector t of the expected time to absorption as t = Nv, 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 s_i. Since we only need the expected time from state $s_1 = (0)$ to absorption, the detection time T_d, is:

$$T_d = v_1 N v \qquad (2)$$

where $v_1 = [1, 0, \ldots, 0]$.

For obtaining the overall overhead (or transmission cost) we need to obtain the number of transmitted messages for each state s_i. During state s_1 no node is in the POSITIVE state. In this state, no messages are transmitted and $m_1 = 0$. The second state s_2 starts when 1 node has a POSITIVE state (that is, there is one sender). In this case, this POSITIVE can be transmitted to all nodes (except itself) for the duration of this state (denoted as f_2) with a rate λ and probability p_c. Then, the expected number of messages can be obtained as $m2 = f_2\lambda(C - 1)p_c$. For state s_3, we have 2 possible senders, so $m_3 = 2f_3\lambda(C - 1)p_c$. Then, for state s_i we have (i − 1) senders, so $m_i = (i - 1)f_i\lambda(C - 1)p_c$. We can obtain the duration of each state 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). Summing up, the cost of transmission is:

$$M_d = \lambda(C - 1)p_c \sum_{i=1} \Phi(s_i) N(1, i) \qquad (3)$$

Where $\Phi(s_i) = (i - 1)$ is the number of senders in state s_i. We can now extend the previous basic model to the case of several selfish nodes (S > 1). The solution is based on using a Continuous Time Markov Chain with S dimensions. We start with S = 2, so we have a two-dimensions CTMC (for short, a 2D-CTMC). Each state s_i now has two values

(1) (c_2, c_1), where c_1 is the number of collaborative nodes having a POSITIVE for selfish node 1, and c_2 is the same for selfish node 2.

At the beginning all nodes are in the NOINFO state. Then, when a contact occurs, c_1 and c_2 can increase by one. The final (absorbing) state is when (c_2, c_1) = (C, C). So, the 2D-CTMC has an initial state $s_1 = (0, 0)$, $s_\tau = (C + 1)^2 - 1$ transient states (from $s_1 = (0, 0)$ to s_τ = (C − 1, C) state) and $\upsilon = 1$ absorbing state $s_{\tau+1}$ = (C, C). Now, we derive the transition rates p_{ij} for the transition matrix. Given the state $s_i = (c_2, c_1)$, the following transitions can occur:

• (c_2, c_1) to (c_2, c_1 + 1): the same that in S = 1 model, replacing c by c_1, $t_{c1} = (\lambda p_d + \lambda p_c c_1)(C - c_1)$

• (c_2, c_1) to (c_2 + 1, c_1): the same for c_2, $t_{c2} = (\lambda p_d + \lambda p_c c_2)(C - c_2)$

• (c, c) to (c, c): $t_0 = 1 - t_{c1} - t_{c2}$

and, using equation 2, we can obtain the detection time (T_d). We can extend this model to the case of S > 2. Then we have $\tau = (C + 1)^S - 1$ transient states and, for each state $s_i = (c_S, c_{S-1}, \ldots c_2, c_1)$, the transition rate from c_j to c_j +1 is $t_{cj} = (\lambda p_d + \lambda p_c c_j)(C - c_j)$.

For the overhead, we assume that a node transmits only one message for all the positives it has. Then, the number of messages in each state depends on the distribution of the positives. Obtaining all the combinations when S is high can be very complex, but a simple approximation based on bounding the value of senders can be used. It is easy to see that the number of senders in each state is between the maximum of c_j and the minimum between the sum of c_j and C.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

That is, $\max(s_i) \leq \Phi(s_i) \leq \min(\text{sum}(s_i), C)$ where $\max(s_i) = \max^{j=1} c_j$ and $\text{sum}(s_i) = \sum^{j} =1\ cj$ Then, we estimate the number of senders $\Phi(s_i)$ by calculating the mean of the lower and upper bounds. Finally, the number of messages is obtained using equation 3

Now, we briefly describe the validation process of the model previously presented. This performance model obtains the time and overhead (T_d, M_d) from the following set of inputs: the rate of contacts (λ), the network (N, C, S) and the watchdog (p_c, p_d) parameters. We used the ns-2 setdest command to create contact traces, which are used, on the one hand to fit the λ value that is used in our performance model and on the other hand to simulate the contacts to obtain the simulation results. We validate our model using a set of 100 random tests. The tests have different parameters values (N, C, S, p_c, p_d) and mobility patterns (mean speed of nodes v, communication range r, side l, etc.). For each test, we repeated the simulation 1000 times in order to obtain values with confidence intervals for the detection time and the overhead. These values are compared with the results of our model in order to obtain the accuracy of our model. After running all the tests we obtained the mean error (and 95% confidence intervals) for T_d and M_d. For the detection time the mean relative error was 2.18% ([0.52, 3.95]) and for the overhead it was 2.86% ([0.77, 6.48]). These results confirm that the error of our model is very low.

IX. Evaluation Results

This section is first devoted to evaluating the performance of our collaborative watchdog using the performance model detailed in section III. All the model was implemented and evaluated using Matlab. For the following evaluations we consider a contact rate of 0.0135 contacts/h, $\lambda_v = 3.71\times10^{-6}s^{-1}$. This value was calculated in [12] based on real motion traces from about 2100 operational taxis for about one month in Shanghai city. The first evaluation shows the influence of the degree of collaboration in a network with 50 nodes and one selfish node (see figure 5a) with different detection probabilities values (p_d). We can see that increasing the degree of collaboration from 0 to 0.2 reduces the detection time exponentially and increases the overhead (cost) exponentially as well. This reduction is quite significant for low detection probabilities (p_d = 0.1). For p_c = 0 (no collaboration), the detection time is 12×10^6s (about 3300 hours). This value can be greatly reduced by using our collaborative watchdog. Thus, if all nodes implement the collaborative approach (p_c = 1) the detection time is reduced to 30 hours.

Even for a low collaboration rate (p_c = 0.2) the time is reduced to 78 hours. For both cases, the overhead is approximately of 210 messages (less than 7 messages by hour, a much reduced cost). We can also see that increasing the probability of collaboration (from 0.4 to 1) has low impact on both the detection time and the overhead, which emphasizes on the resilience of our collaborative approach. The second evaluation shows the impact of the number of nodes ranging from 0 to 100 (see figure 5b). Three different sets of values for p_c and p_d were used.

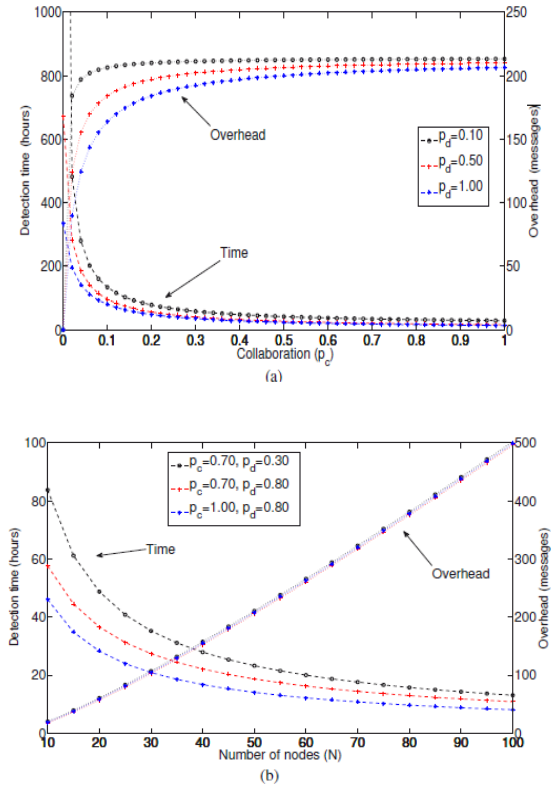

Fig. 5: Evaluation for S = 1. a) depending on pc b) depending on N.

The first set (1, 0.8) is a full collaborative network with a high probability of detection, the second set has a reduced degree of collaboration (0.7), and finally the last set has a low probability of detection (0.3). We observe that, in general, the greater the number of nodes, the lesser the detection time and the greater the number of messages. As expected, reduced values of collaboration and detection probabilities imply greater detection times.

Figure 6 shows the influence of the number of selfish nodes S for N = 50. As expected, the detection time increases when the number of selfish nodes is higher.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Regarding the overhead, we can see that the number of messages increases exponentially for low values of S, and then it decreases slowly, for S > 10. The reason is that, when the number of selfish nodes is high, the collaborative nodes are reduced and they can transmit fewer messages.

More experiments were performed using different λ values, for example with a contact rate of 0.101 contacts/h, obtained from human mobility traces [7], and the results obtained were similar to those presented here.

Now we proceed to compare our collaborative watchdog approach with previous cooperative approaches that use periodic messages for the diffusion of information about positives detections. If a node has information about a positive, 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 simulations. We implemented the periodic diffusion protocol, as described in the previous paragraph. By using the ns-2 setdest command we generate mobility scenarios that are used to simulate both approaches.

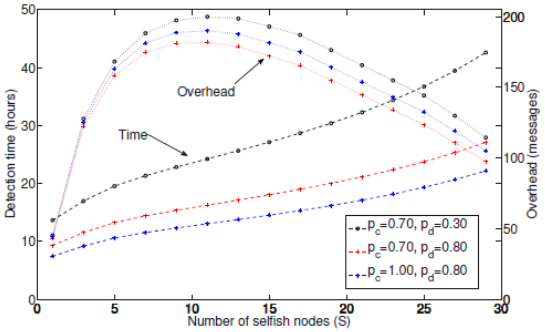

Fig. 6: Evaluation depending on S for N = 50

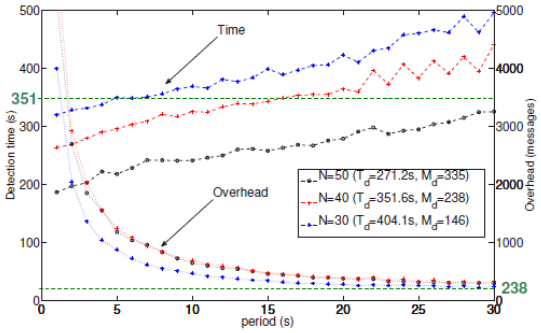

Fig 7

Fig. 7: Detection time and overhead depending on period P for the periodic approach. The main parameters for the mobility model are mean-speed = 5m/s, side-area = 1000 m, pause-interval = 1s, range = 100m

Figure 7 shows the detection time and overhead for P ranging from 1 to 30s for the periodic diffusion protocol with three different number of nodes. The results confirm that increasing the period P implies that the detection time is increased and the overhead reduced. We can compare these results with the detection time and overhead values for our collaborative watchdog (that are in the legend of the plot). For example, for N = 40, the periodic diffusion for periods below 15s has a shorter detection time than our model but with a higher overhead. For example, for P = 5s, the detection time is 295s (a reduction of 15%) and the overhead is 1253 messages (an increment of 526%). For P = 15s, the detection time is similar to our approach, and the overhead is 483 messages (205% higher). We conclude that, although using periodic diffusion can reduce the detection time slightly, this implies a large overhead.

X. CONCLUSIONS

In this paper we have proposed and evaluated a new collaborative watchdog approach. We modelled its performance using a Continuous Time Markov Chain with two parameters to indicate the degree of collaboration and detection of the watchdog.

Numerical results show that a collaborative watchdog can reduce the overall detection time with a reduced cost in term of message overhead. This reduction is very significant when the watchdog detection effectiveness is low. Furthermore, this reduction can be obtained even with a moderate degree of collaboration.

As future work, we plan to extend this model to evaluate the effect of false positives and false negatives. Such extension poses several problems: first, a node needs to transmit not only the positives but also the negatives, so it will increase the overhead; second, when a node receives this information about positives and negatives, conflicts with previous information may appear (for example, when a node has a negative about a given node and it receives a positive). So, an updating strategy may be needed. We also plan to evaluate the case of malicious or cheating behaviour by introducing some kind of reputation scheme. Finally, we are also planning to implement this collaborative watchdog in a test bed

REFERENCES

[1] J. Hotelman, J. C. Ruiz, and P. Manzoni, "Evaluating the uselfusness of watchdogs for intrusion detection in VANETs," in 2010 ICC Workshop on Vehicular Networking and Applications.

[2] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proc. 2000 MobiCom, pp. 255-265.

[3] K. Paul and D. Westhoff, "Context aware detection of selfish nodes in DSR based ad-hoc networks," in Proc. 2002 IEEE Globecom.

[4] S. Buchegger and J.-Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems," IEEE Commun. Mag., vol. 43, no. 7, pp. 101-107, July 2005.

[5] M. Karaliopoulos, "Assessing the vulnerability of DTN data relaying schemes to node selfishness," IEEE Commun. Lett., vol. 13, no. 12, pp. 923-925, Dec. 2009.

[6] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, "Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks," IEEE Commun. Lett., vol. 14, no. 11, pp. 1026-1028, Nov. 2010.

[7] Y. Li, G. Su, D. Wu, D. Jin, L. Su, and L. Zeng, "The impact of node selfishness on multicasting in delay tolerant networks," IEEE Trans. Veh. Technol., vol. 60, no. 5, pp. 2224-2238, June 2011.

[8] H. Otrok, M. Debbabi, C. Assi, and P. Bhattacharya, "A cooperative approach for analyzing intrusions in mobile ad hoc networks," in Proc. 2007 International Conference on Distributed Computing Systems Workshops, p. 86.

[9] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," Performance Evaluation, vol. 62, pp. 210-228, Oct 2005.

[10] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in Proc. 2007 MobiCom, pp. 183-194.

[11] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in Proc. 2009 MobiHoc, pp. 299-308.