Collaborative Watchdog for Treating Selfish Nodes in MANET

ABDUL MAJEED KHAN (amkhan_08@yahoo.com) ,PG Scholar, S.A.ENGINEERING COLLEGE, CHENNAI
PRIYA.S(sakthipriya@gmail.com), Assistant Professor, S.A.ENGINEERING COLLEGE, CHENNAI.

ABSTRACTA
wireless heterogeneous sensor network has more number of sensor nodes with constrained power and computational difficulties along with supernodes with deficient energy resources is supported by disjoint path vector algorithm also known as distributed fault tolerant topology control algorithm.The main issue of N degree anycast topology control is found in DPV mechanism,where the aim is to designate each sensor's coverage area so as to reach the nodeswith minimum k-vertex disjoint paths to the supernodes.This aids in reducing the total energy consumption.While restricting N-1 node failures,this topology becomes fault tolerant and maintains better connectivity.In restriction of N-1 node failures,the resulting topologies tolerate node failures in the worst case. High Quality of Service network is always appreciable in any network. A path information collection is done through DPV algorithm which also considersQoSfor heterogeneous WSN.

Keywords: fault tolerance, k-connectivity, Topologycontrol, disjoint paths, heterogeneous wireless sensor network,Qos.

I.Introduction

Mobile Ad Hoc Network (MANET) is a group of mobile nodes (hosts) which correspond with each other via wireless associations either directly or depending on other nodes as routers. The function of MANETs does not depend on already existing backbone infrastructure or base stations. Network nodes in MANETs are liberated to move haphazardly. Hence forth, the hierarchal order of the network topology of a MANET may alter swiftly and impulsively. All network behavior, such as identifying and discovering the topology and transportation of data packets, have to be carried out by the nodes themselves, either in isolation or cooperatively in a group. Relaying on its purpose and function, the configuration of a MANET may contrast from a tiny, stationary network that isextremely power-constrained to a outsized range, movable or mobile, exceedingly vibrant network.

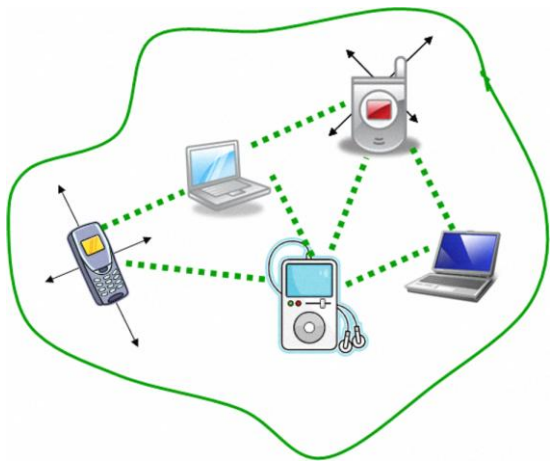


Fig.1 MANET illustration

There are two types of MANETs: closed and open [1]. In a closed MANET, all mobile nodes collaborate with each other toward a widespread objective. In an open MANET, dissimilar mobile nodes with diverse objectives contribute to their resources in order to achieve universal connectivity.

Nevertheless, quantities of resources are consumed rapidly as the nodes take part in the network functions and activities. For example, battery power is considered to be most significant in a mobile atmosphere and environment. An isolated or individual mobile node may try to take advantage from other nodes, but decline to contribute its own resources. To be precise such a conditional nodes are called selfish or misbehaving nodes and their attribute is termed selfishness or misbehavior [2]. One of the foremost sources of energy utilization in the mobile nodes of MANETs is wireless transmission [3]. A selfish node refuses to forward data packets for other nodes in order to conserve its own energy.
In [II], the authors inspect the brunt of selfish nodes. They viewed selfish nodes as nodes that took gain of other nodes for their self communication requirement. They projected identification methods based on: (a) listening for passive (inert) acknowledgment or response, and (b) timer to trace on unreasonable delays. Their countermeasure method is to separate the selfish node. Selfish assessment was made by means of the throughput, false detection rate, and chance of connection deprivation metrics. The authors also highlighted the severe cost of fake detection, i.e., mistaking unselfish nodes for selfish nodes.

This paper introduces Collaborative Watchdog for detecting selfish nodes that combines local watchdog detections and the propagation of this information on the network. By chance one of the nodes has beforehand identified a selfish node it can broadcast this information to other nodes when a communication occurs. By this mechanism, nodes possess back-end information concerning the selfish nodes in the network configuration. The objective of our approach is to minimize the detection time and to get better the accuracy, also the precision by the way.

A misbehaving node can ultimately transform to be healthy one. How can one handle with such a changeover? Hence, in a way analogous to human's personality, we deem that selfishness can also amend with time, place, environment, circumstances, status etc. in abridgment, we outlook that selfishness should be considered as a inconsistent variable, not a constant. Selfishness is a time-varying condition-dependent state. Precise designing of selfishness is essential to fully comprehend its force on ad hoc network process and communication efficiency.

2. RELATED WORK

The Effect of node selfishness on MANETs has been studied in [30], [31], [32]. In [32] it is shown that when no selfishness prevention method is present, the packet delivery rates turn out to be criticallydegraded, from a rate of 80 percent when the selfish node ratio is 0, to 30 percent when the selfish node ratio is 50 percent. In DTNs, selfish nodes can severely degrade the efficiency of packet transmission. For instance, in two-hop relay schemes, if a packet is transmitted to a selfish node, the packet is not reflected, therefore being missed. Therefore, isolating such nodes rapidly and precisely is indispensable for the on the whole performance of the network. Preceding works have confirmed that watchdogs are appropriate device to detect misbehaving and selfish nodes.

Fundamentally, watchdog systems listen in wireless traffic and scrutinize it to decide whether neighbor nodes are behaving in a selfish mode [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). However, watchdogs can be unsuccessful on this detection, generating false positives and false negatives that dangerously degrade the behavior of the system on whole.

The detection and elimination approach is a clear-cut way to deal with with selfish nodes and numerous solutions have been offered [3], [14], [19], [22], [23], [24], [25], [28], [34]. In previous works it has been shown how some degree of collaboration 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 neighbors to firmly detect misbehaving nodes. The system's retort is to isolate those nodes from the network, punishing then for the foreseeable future. A distributed intrusion detection system (IDS) is introduced in [35]. In this approach if a node in the vicinity detects an intrusion with strong verification, it can commence a response. Nevertheless, if a node detects an irregularity with weak proof, it can initiate a supportive universal intrusion detection process. A comparable approach is the mobile intrusion detection system described in [20]. In this case, local sensor ratings are occasionally flooded all over the network in order to acquire a universal rating for each misbehaving node. A further approach is CORE "collaborative reputation mechanism" [26]. The CORE system is like the distributed IDS approaches described below. It consists in local examination using watchdogs that are pooled and disseminated to obtain a status (reputation) for each node. This reputation is used to resolve whether a node is permitted to partake (otherwise, it is excluded). Another approach is OCEAN [2] where the reputation of a neighbor is evaluated by means of locally available information, avoiding composite and potentially susceptible techniques of reputation broadcast throughout the network. It is shown that, even with direct neighbor observations, OCEAN performs almost as well as those plans that contribute to back-end reputation information. In [14] ansystematic selfish model (which is tied specifically to the Ad hoc on-demand distance vector (AODV) routing protocol) is proposed. A recent work [34], initiates the audit-based misbehavior detection (AMD) which segregates continuous and selective packet droppers. The AMD system incorporates reputation management, reliable route discovery, and recognition of misbehaving nodes based on behavioral audits. This method also gathers first and second-hand information for obtaining the reputation of nodes.

More recently, papers have emphasized on DTNs. In [19], the author introduces a replica for DTN data relaying schemes under the impact of node selfishness. A analogous approach is presented in [23] that shows the effect of collectively (social) selfish behavior. Social selfishness is an addition of classical selfishness (also called individual selfishness). A social selfish node can collaborate with other nodes of the same cluster, and it does not work together with other nodes outside the cluster. 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]. However, these approaches do not assess the consequence of false positives, false negatives and malicious nodes. For instance, the approach in [26] only transmits positive detections. The setback, as shown in the estimation sections, is that if a false positive is produced it can broadcast this wrong information very swiftly on the network, separating nodes that are not selfish. Hence, an approach that comprises the dissemination of negative detections as well becomes required.

Another setback is the impact of colluding or malicious nodes. Even though a reputation system, as the one presented in [26], can be helpful to alleviate the effect of malicious nodes, it obviously depends on how are joint local and global ratings, as shown in this paper.

Another performance issue is the high forced overhead due to the flooding process in order to attain a fast diffusion of the information. in view of the fact that our approach is based on contacts, it has been verified that the overhead is greatly reduced.

3. Architecture model

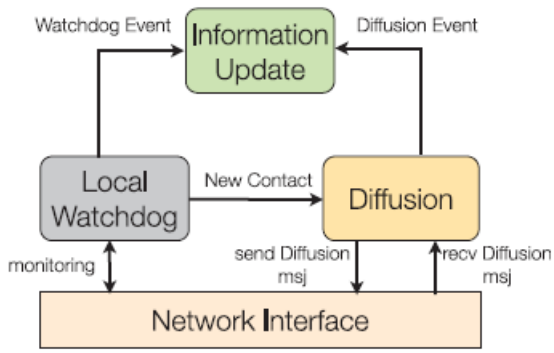


Fig. 2 shows the functional structure of CWS

Now Let us see in detail its three main components.

The Local Watchdog has two purposes: the detection of selfish nodes and the detection of new contacts. The local watchdog can create the following events about neighbor nodes: PosEvt (positive event) when the watchdog isolates a selfish node, NegEvt (negative event) when the watchdog isolates that a node is not selfish, and NoDetEvt (no detection event) when the watchdog does not have adequate information about a node (for example if the contact time is very less or it does not listen enough responses). The detection of new contacts is based on neighborhood packet listening; thus, when the watchdog listens packets from a new node it is understood to be a new contact, and so it makes an event to the network information component.

The Diffusion module has two roles: the transmission plus the reception of positive (and negative) detections. A vital issue of our approach is the diffusion of information. As the number of selfish nodes is low contrast to the total number of nodes, positive detections can constantly be transmitted with a low overhead. Nonetheless, transmitting only positive detections has a serious downside: false positives can be broadcasted over the network very fast. Thus, the transmission of negative detections is essential to counterbalance the effect of these false positives, but transferring all known negative detections can be troublesome, producing extreme messaging or the fast diffusion of false negatives. Thus, we commence a negative diffusion factor 'g' 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 assessment section that a low value for the g factor is sufficient to neutralize the effect of false positives and false negatives. In conclusion, when the diffusion unit receives a new contact event from the watchdog, it transmits a message together with this information to the new neighbor node. While the neighbor node receives a message, it produces an event to the network information module with the record of these positive (and negative) detections.

Informing or consolidating the information is another key issue. This is the job of the Information Update module. A node can have the subsequent internal information about other nodes: NoInfo state, Positive state and Negative state. A NoInfo state understood to be that it has no information about a node, a Positive state means it believes that a node is selfish, and a Negative state assured and believes that a node is not selfish. A node can have direct information (from the local watchdog) and indirect information (from neighbour nodes). CWS 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 exacting, these events updates a reputation value r using the following expression:

$$\rho = \rho + \Delta$$

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

In general, a PosEvt event increases the reputation value while a NegEvt event decreases it. Defining 'u' as a threshold and using the reputation value r, the state of the node modifies to Positive if $\rho > u$, and to Negative if $\rho < -u$. Or else, the state is NoInfo. The arrangement of d and u parameters allows a very flexible and vibrant behavior. First, if $u > 1$ and $\rho < u$ we need several events in order to change the state. For example, starting from the No Info state, if $u = 2$ and $\delta = 1$, at least a local and an indirect event is needed to change the state, but if $u = 1$, 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 $u = 3$, means that we need one local event and one indirect event, or three indirect events, to change the state. This approach can give back wrong local decisions: for example, a local NegEvt can be compensated by $2\ \delta + u$ indirect PosEvt events, and in order to change from Positive to Negative states (or vice-versa) we need twice the events.

The merits of this updating strategy are twofold. First, with the threshold u 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.

To end with, the network information about the nodes has a termination time, so after some time devoid of contacts it is updated. The accomplishment of this method is clear-cut. When an event is established, it is marked with a time stamp, so in a given timeout a conflicting event is generated, in order to update the value of δ

3.1 The Model for the Detection of Selfish Nodes

The objective is to obtain the detection time (and overhead) of a selfish node in a network. This model takes into consideration the impact of false negatives. False positives do not influence the detection time of the selfish node, so Pfp is not introduced in this model. Using (lambda) as the contact rate between nodes, we can model the network using a 4D continuous time Markov chain (4DCTMC). For modeling reason, the communicating nodes are divided into two sets: a set with D destination nodes and a set of E = C _ D intermediate nodes.

where T is a t x t matrix with elements qij denoting the transition rate from transient state si to transient state sj, R is a t x y matrix with elements qij denoting the transition rate from transient state si to the absorbing state sj, the left 0 is a y x t zero matrix, and the right 0 is a y x y zero matrix.Beginning with the detection time, from the 4D-CTMC we can infer how much time it will take for the process to be carried out. 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 one's (v = [1, 1 . . . 1] ^T). the detection time Td, is:

Where T is a random variable denotingthe detection time for all nodes and the overhead of transmission (number of messages) is:

Using the generator matrix Q we can get two different expressions: one for the detection time Td and another for the complete overhead (or cost) Od.

3.2 Algorithm for detecting selfish nodes

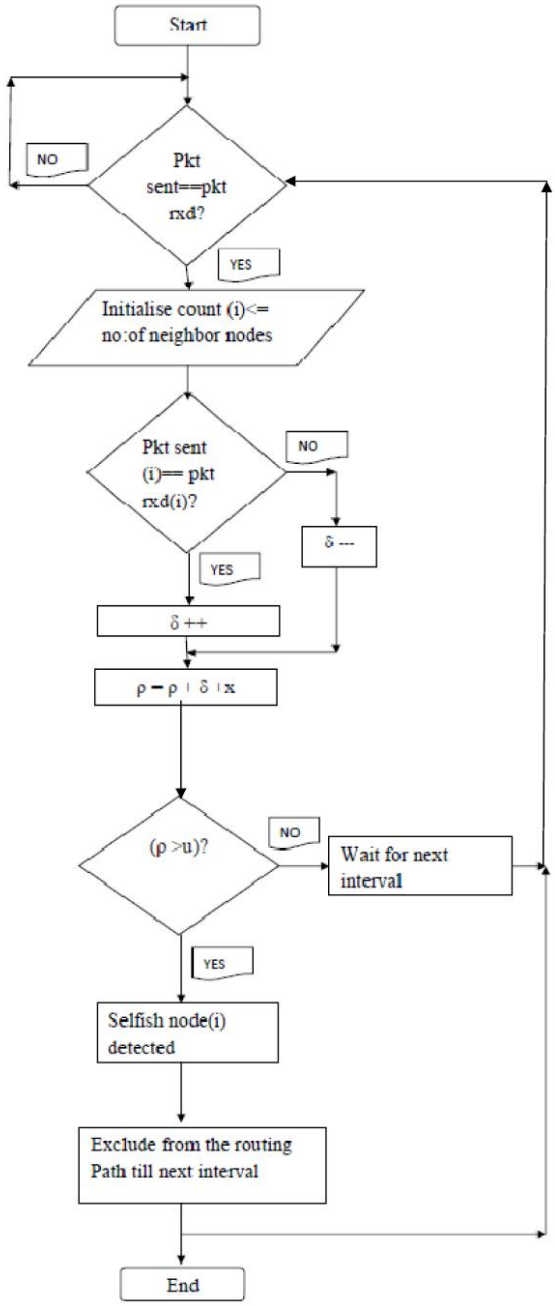
The algorithm process as follows:-
The source stays for the destination to send acknowledgement to it after every packet. If source receives the acknowledgement, then there is no misbehavior in the network and process continues as such. But if the destination fails to acknowledge the data packets for a time period, then IDS starts its processes. The proposed IDS algorithm manipulates the success rate for each node and compares it with the threshold (base) value. If lesser than the threshold, corresponding node is marked as selfish and values δ=1. This value added together with the reputation value of neighbor node about the focus node. The total value of ρ if exceeds The total value of ρ if exceeds u marked in routing table as Selfish node until the status is changed after the next upgrade.

Input: Threshold_value, Set_of_neighbour_nodes; Set_of_nodes_who_sent_route_reply; source; destination.
Begin
If(pkt_received_by_dest==pkt_sent_by_source) then
network does not shows any misbehaviour
Else if(pkt_received_by_dest< certain percentage of pkt sent by source over the network)
{
Then the network shows misbehaviour
For(inti=0; i<no._of_nodes_who_sent_route_reply; i++) 0
{
If(no.ofpkt sent to node(i)==no.ofpktrxd from node(i)) then
δ =1
Else
δ =-1
ρ = ρ + δ +x
if (ρ >u)
result = node(i) is a selfish node
return result
END if
End else if
EXIT

This can be illustrated in terms of flowchart as:

The source waits for the destination to send acknowledgement to it after every packet. If transmitter,receives the acknowledgement, then there is no misbehavior in the network and process proceeds as such as show in flowchart. But if the destination fails to acknowledge the data packets for a length of period, then IDS starts its processes as illustrated in the flowchart loop.. The proposed IDS algorithm manipulates the success rate for each node and compares it with the threshold (base) value. If lesser than the threshold, corresponding node is marked as selfish and values δ=1. This value added together with the reputation value of neighbor node about the focus node. The total value of ρ if exceeds The total value of ρ if exceeds u marked in routing table as Selfish node until the status is changed after the next upgrade

Owing to the effects of the RN's intrinsic and extrinsic factors on its node-selfishness of forwarding multi-services, it should quantify these effects, thus the models of the intrinsic and extrinsic selfishness are designed By instinct, the individual network nodes would prefer to act selfishly rather than altruistically in distributed network secaniros.



Flowchart for the Algorithm

4. PERFORMANCE EVALUATION

During simulation time, the trace file is used to determine the events occurred in that simulation environment. The executed

trace files are used to analyze the behavior of the network. While execution all events are recorded and stored .During this process, we record parameters like packet delivery rate, packet lost rate, delay, throughput, residual energy etc., of the different schemes considered in the proposed work. These readings are stored in the trace files.

4.1EXPERIMENTAL SETUP

The Simulation is carried out using the tool Network Simulator 2 (NS-2) version 2.35 on Linux operating system Ubuntu version 12.10.The system runs on a laptop with Core 2 Duo T6500 processor with 4-GB RAM. For plotting graph, trace-graph version 202 is used.

Number of Nodes: 10 of which 5 were communicating
Packet traffic: CBR (Constant bit rate) on UDP
Packet Size: 512B
Packet Interval: 0.25
Routing Protocol: AODV

The simulation result shows the detection of selfish nodes with the help of CWS.

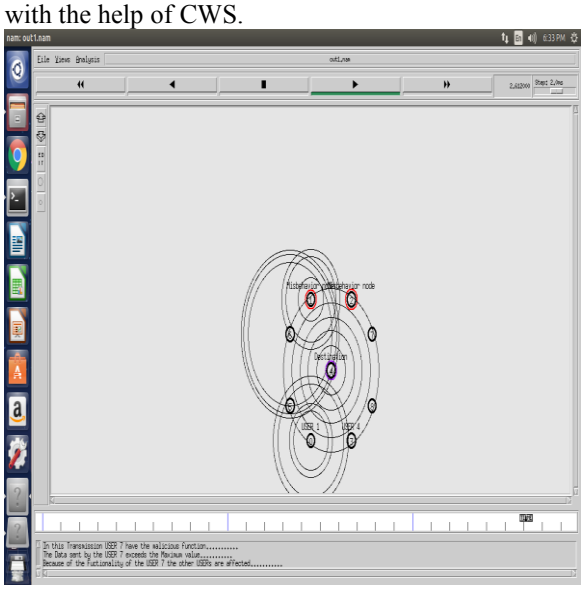


***Fig.3*: Packet Transmission between source & Destination**

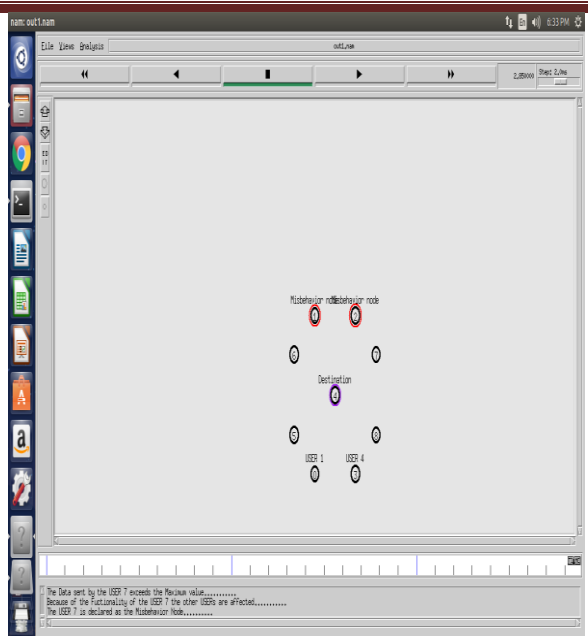


Fig.4: Selfish-node Detection

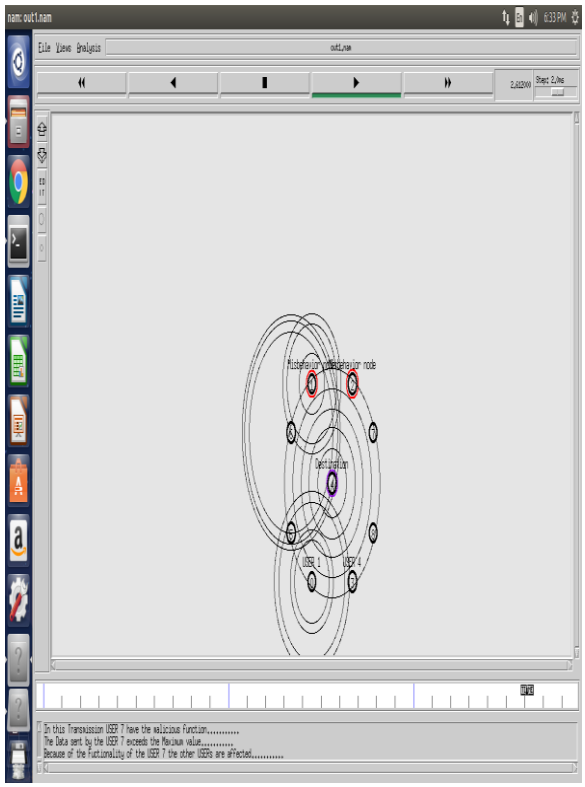


Fig.5: Transmission of packets bypassing the Selfish-nodes after detection

First, we plot the average detection delay with the existence of selfish nodes. The below graph is plotted for the analytical and simulation results.

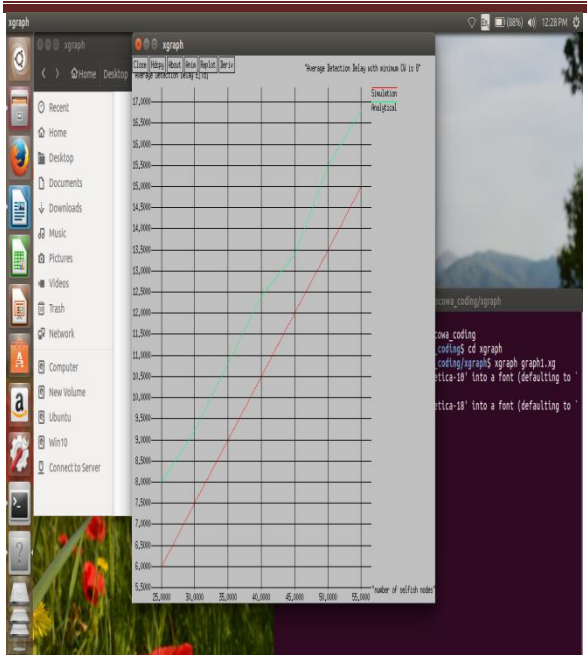


Fig.6: Average Detection Delay in MANET Scenario with selfish-nodes

The second analysis is the impact of number of nodes to the detection time of CWS. Where the below graph shows that the detection time is considerably reduced when the network is more populated with malicious node.

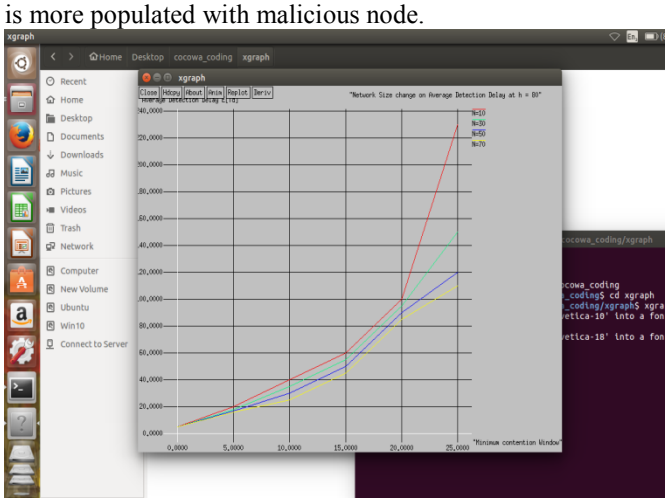


Fig.7: Detection Time of CWS with impact of No. of Nodes

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 informationabout the positive detection is disseminated more quickly. The cost is directly proportional to N.

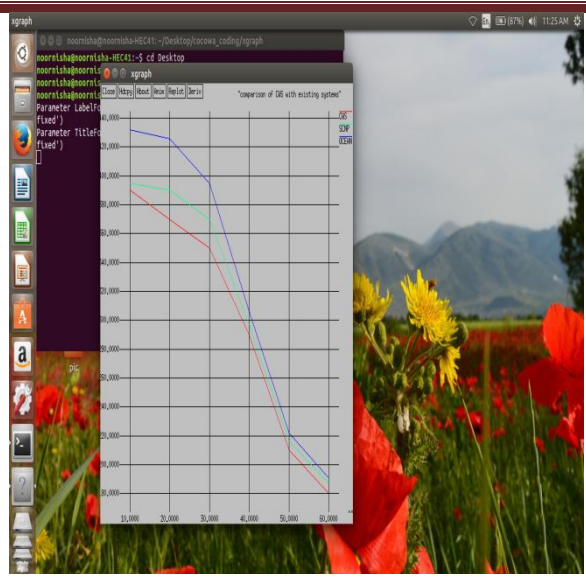


Fig.8: Comparison of CWS with Existing System

Finally, we compare our approach to the classic periodic diffusion model, where we can see that the detection time is greatly reduced even for low values, so CWS 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. 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.

CONCLUSION

This paper proposes CWS as a 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. CWS 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 CWS 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 percent for very low degree of collaboration to 99 percent 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 CWS we can reduce the effect of malicious or collusive nodes. If malicious nodes spread false negatives or false positives in the network CWS is able to reduce the effect of these malicious nodes quickly and effectively. Additionally, we have shown that CWS is also effective in opportunistic net- works 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.

REFERENCE

[1] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat, "Lightweight sybil attack detection in manets," IEEE Syst. J.,vol. 7, no. 2, pp. 236–248, Jun. 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," IEEE Commun. Mag., vol. 43, no. 7, pp. 101–107, Jul. 2005.

[4] L. Butty_an and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in Proc..Workshop Mobile Ad Hoc Netw.Comput., 2000, pp. 87–96.

[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. Lett., vol. 16, no. 5, pp. 642–645, May 2012.

[6] M. Hollick, J. Schmitt, C. Seipl, and R. Steinmetz, "On the effect of node misbehavior in ad hoc networks," in Proc. IEEE Int. Conf. Commun., 2004, pp. 3759–3763.

[7] C. K. N. Shailender Gupta and C. Singla, "Impact of selfish node concentration in MANETs," Int. J. Wireless Mobile Network., vol. 3, no. 2, pp. 29–37, Apr. 2011.

[8] C. Toh, D. Kim, S. Oh, and H. Yoo, "The controversy of selfish nodes in ad hoc networks," in Proc. Adv. Commun. Technology,
Feb. 2010, vol. 2, pp. 1087–1092.

[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 Trans. Mobile Comput., vol. 7, no. 1,pp. 19–33, Jan. 2008.

[10] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in Proc. 10th ACM Int. Symp. Mobile Ad Hoc Netw.Comput., 2009, pp. 299–308.

[11] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," Perform. Eval., vol. 62, pp. 210–228, Oct. 2005.

[12] 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. Lett., vol. 16, no. 5, pp. 642–645, May 2012.

[13] E. Hern_andez-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 Proc. 15th ACM Int.Conf. Modeling, Anal.Simul. Wireless Mobile Syst., New York, NY, USA, 2012, pp. 159–166.

[14] M. Hollick, J. Schmitt, C. Seipl, and R. Steinmetz, "On the effect of
node misbehavior in ad hoc networks," in Proc. IEEE Int. Conf. Commun., 2004, pp. 3759–3763.

[15] J. Hortelano, J.-C. Cano, C. T. Calafate, M. de Leoni, P. Manzoni, and M. Mecella, "Black hole attacks in p2p mobile networks dis- covered through Bayesian filters," in Proc. Int. Conf. Move Mean- ingful Internet Syst., 2010, pp. 543–552.

[16] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the useful- ness of watchdogs for intrusion detection in VANETs," in Proc. Int. Conf. Commun. Workshop, 2010, pp. 1–5.

[17] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based for- warding in delay tolerant networks," in Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw.Comput., 2008, pp. 241–250.

[18] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovi,c, "Power law and
exponential decay of inter contact times between mobile devices," in Proc. ACM MobicomAnnu. Int. Conf. Mobile Comput.Netw., 2007, pp. 183–194.

[19] 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.

[20] F. Kargl, A. Klenk, S. Schlott, and M. Weber, "Advanced detection of selfish or malicious nodes in ad hoc networks," in Proc. 1st Eur. Conf. Security Ad-Hoc Sens. Netw., 2004, pp. 152–165.

[21] F. Kargl, A. Klenk, M. Weber, and S. Schlott, "Sensors for detection of misbehaving nodes in MANETs," in Proc. Detection Intru- sions Malware Vulnerability Assessment, 2004, pp. 83–97.

[22] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay toler- ant networks," in Proc. IEEE Conf. Comput.Commun, 2010, pp. 857–865.

[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," IEEE Trans. Veh. Technol., vol. 60, no. 5, pp. 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 Trans. Mobile Comput., vol. 10, no. 7, pp. 997–1010, Jul. 2011.

[25] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing mis- behavior in mobile ad hoc networks," in Proc. ACM MobicomAnnu. Int. Conf. Mobile Comput.Netw., 2000, pp. 255–265.

[26] P. Michiardi and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in Proc. 6th Joint Working Conf. Commun. Multimedia Secur., 2002, pp. 107–121.

[27] A. Passarella, and M. Conti, "Characterising aggregate inter- contact times in heterogeneous opportunistic networks," in Proc. 10th Int. IFIP TC 6 Conf. Netw., 2011, pp. 301–313.

[28] K. Paul and D. Westhoff, "Context aware detection of selfish nodes in DSR based ad-hoc networks," in Proc. IEEE Global Tele- commun. Conf., 2002, pp. 178–182.

[29] M. D. Serrat-Olmos, E. Hern,andez-Orallo, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A collaborative Bayesian watchdog for detecting black holes in MANETs," in Proc. 6th Int. Symp. Intell.Distrib.Comput. VI, 2012, vol. 446, pp. 221–230.

[30] C. K. N. Shailender Gupta and C. Singla, "Impact of selfish node concentration in MANETs," Int. J. Wireless Mobile Netw., vol. 3, no. 2, pp. 29–37, Apr. 2011.

[31] C. Toh, D. Kim, S. Oh, and H. Yoo, "The controversy of selfish nodes in ad hoc networks," in Proc.Adv.Commun. Technol., Feb. 2010, vol. 2, pp. 1087–1092.

[32] Y. Yoo, S. Ahn, and D. Agrawal, "A credit-payment scheme for packet forwarding fairness in mobile ad hoc networks," in Proc. IEEE Int. Conf. Commun., May 2005, vol. 5, pp. 3005–3009.

[33] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," Comput.Netw., vol. 51, no. 10, pp. 2867–2891, 2007.

[34] Y. Zhang, L. Lazos, and W. Kozma, "AMD: Audit-based misbe- havior detection in wireless ad hoc networks," IEEE Trans. Mobile Comput., vol. PP, no. 99, 2012, http://doi.ieeecomputersociety. org/10.1109/TMC.2012.257

[35] Y. Zhang, W. Lee, and Y.-A.Huang, "Intrusion detection techniques for mobile wireless networks," Wireless Netw., vol. 9, no. 5, pp. 545–556, Sep. 2003.

[36] S. Zhong, J. Chen, and Y. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in Proc. IEEE

[37] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni, "Recognizing exponential inter-contact time in VANETs," in Proc. IEEE Conf. Comput.Commun., 2010, pp. 101–105.