Detection of Node Activity and Selfish & Malicious Behavioral Patterns using Watchdog – Chord Monitoring

M. Kanimozhi [1], Dr. S. P. Rajagopalan [2].
Department of Computer Science and Engineering
GKM College of Engineering and Technology.
Chennai

***Abstract*- In the node could have a selfish behavior, being unwilling to forward packets for others. So the overall network performance could be seriously affected. In the data nodes are distributed among the cluster. Every node assigned with set of friend and enemies by themselves. There are selfish nodes and normal nodes in the cluster. Selfish node will not transmit packet to enemy nodes. It transmitted data packet only to friends list. Selfish nodes are greedy in transmission. So that it accepts most of the data transmission by themselves in the network. However, the detection process performed by watchdogs can fail, generating false positives and false negative that can induce to wrong operation. We add up another group called malicious. In the malicious node will drop the packet or transmit the packet to the wrong destination. Event it would add up extra data during transmission. We apply chord algorithm to identify behavior pattern of one shelf by two neighborhood nodes and themselves. Servers will finally categories nature of node.**

Keyword-Chord algorithm, selfish node, delay tolerant networks

I.INTRODUCTION

Cooperative networking is currently receiving significant attention as an emerging network design strategy for future mobile wireless networks. Successful cooperative networking can prompt the development of advanced wireless networks to cost-effectively provide services and applications in contexts such as vehicular ad hoc networks (VANETs) or mobile social networks. Two of the basic technologies that are considered as the core for these types of networks are mobile ad-hoc networks (MANETs) and opportunistic and delay tolerant networks (DTNs). The cooperation on these networks is usually contact based. Mobile nodes can directly communicate with each other if a contact occurs (that is, if they are within communication range). Supporting this cooperation is a cost intensive activity for mobile nodes. Thus, in the real world, nodes could have a selfish behavior, being unwilling to forward packets for others. Selfishness means that some nodes refuse to forward other nodes' packets to save their own resources. The literature provides two main strategies to deal with selfish behaviour: a) motivation or incentive based approaches, and b) detection and exclusion. The first approach, tries to motivate nodes to actively participate in the forwarding activities. These approaches are usually based on virtual currency and/or game theory models. The detection and exclusion approach is a straight-forward way to cope with selfish nodes and several solutions have been presented.

In CoCoWa, we do not attempt to implement any strategy to exclude selfish nodes or to incentivize their participation; instead, we focus on the detection of selfish nodes. The impact of node selfishness on MANETs has been studied in. In it is shown that when no selfishness prevention mechanism is present, the packet delivery rates become seriously degraded, from a rate of 80 percent when the selfish node ratio is 0, to 30 percent when the selfish node ratio is 50 percent. The survey shows similar results: the number of packet losses is increased by 500 percent when the selfish node ratio increases from 0 to 40 percent. A more detailed study shows that a moderate concentration of node selfishness (starting from a 20 percent level) has a huge impact on the overall performance of MANETs, such as the average hop count, the number of packets dropped, the offered throughput, and the probability of reachability. In DTNs, selfish nodes can seriously degrade the performance of packet transmission. For example, in two-hop relay schemes, if a packet is transmitted to a selfish node, the packet is not re-transmitted, therefore being lost. Therefore, detecting such nodes quickly and accurately is essential for the overall performance of the network. Previous works have demonstrated that watchdogs are appropriate mechanisms to detect misbehaving and selfish nodes. Essentially, watchdog systems overhear wireless traffic and analyse it to decide whether neighbour nodes are behaving in a selfish manner. 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 behavior of the system.

II. RELATEDWORK

In the node could have a selfish behavior, being unwilling to forward packets for others. So the overall network performance could be seriously affected. 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 retransmitted [15]. By using this technique, the local watchdog can generate a positive (or negative) detection in case the node is acting selfishly (or not). An example of how CoCoWa works is outlined. It is based on the combination of a local watchdog and the diffusion of information when contacts between pairs of nodes occurs. A contact is defined as an opportunity of transmission between a pair of nodes (that is, two nodes have enough time to communicate between them). Assuming that there is only one selfish node, the figure shows how initially no node has information about the selfish node. When a node detects a selfish node using its watchdog, it is marked as a positive, and if it is detected as a non-selfish node, it is marked as a negative. Later on, when this node contacts another node, it can transmit this information to it; so, from that moment on, both nodes store information about this positive (or negative) detections. Therefore, a node can become aware about selfish nodes directly (using its watchdog) or indirectly, through the collaborative transmission of information that is provided by other nodes. There are two main strategies to deal with selfish behavior in cooperative networks. The first approach tries to motivate the nodes to actively participate in the forwarding activities. For example, in [4], [5] the authors presented a method using a virtual currency called nuglet. In previous works it has been shown how some degree of cooperation can improve the detection of selfish or misbehaving nodes

III. PROPOSEDWORK

In the proposed system, data nodes are distributed among the cluster. Every node assigned with set of friend and enemies by themselves. There are selfish nodes and normal nodes in the cluster. Selfish node will not transmit packet to enemy nodes. It transmitted data packet only to friends list. Selfish nodes are greedy in transmission. So that it accepts most of the data transmission by themselves in the network. However, the detection process performed by watchdogs can fail, generating false positives and false negative that can induce to wrong operation.

1.Network Construction

To implement the Project concept, first we have to construct a network which consists of ‘n’ number of Nodes. So that nodes can request data from other nodes in the network. We will construct the multiple network for our implementation. So that These Networks will have multiple Nodes.

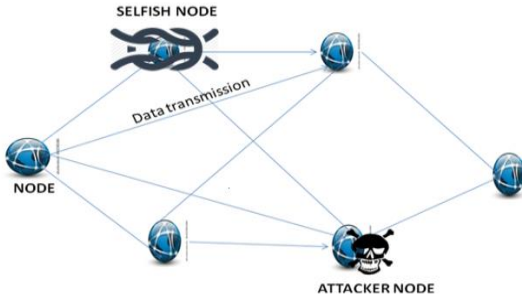

Fig 1: Architecture of Selfish node Detection

2.Friend List Add On

In this module, each node we have to create a Node Frame which contains the Node information, Destination Node field to transfer the data and the browse button to upload the data from Node’s directory. Every node in network has neighbor node details for path selection and communication purpose. Each node in network has friend list.

3.Selfish Node Activity

In this module, every node will add their friend’s names in their list. Every node will spend some energy while transmitting the data in the network. This type of node will try to send the data only to its friend list nodes. If Node 1 adds Node 5 as its friend, node 1 will send the data if it transmits the data via its friend node 5 and it will not transmit the data via some other nodes. These types of nodes are named as Selfish Nodes. Due to the selfish nature and energy consuming, selfish nodes will not transmit the data to some other nodes which are not in their friends list.

4.Malicious Node Activity

As an adversary, the malicious nodes arbitrarily drop others’ bundles (black hole or grey hole attack), which often take place beyond others’ observation in a sparse network, leading to serious performance degradation. These types of nodes will drop the packets or divert the packets to some other nodes which are not the destination nodes. Usually malicious nodes are attacked by the attackers in a network. By this manner normal node is turned down as malicious node.

5.Watch Dog Activity Monitoring

In this module, every node selfish or malicious behavior is monitored and 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 these positive (or negative) detections. Therefore, a node can become aware about selfish nodes directly (using its watchdog) or indirectly. Our main logic is to identify the best nodes through the watchdog activity monitoring.

6.Chord Algorithm

In this module we can verify the Neighbor nodes information of the Requested Node. So that by verifying the Id’s and location we can detect the Clone Node. For this purpose, we have to create the List of the Neighbor Nodes information for each node so that the Server/ can verify the nodes request.

7.Best Route Identification

This is final module of the project. Best Route Identification is one of the important model. Because its identify best way to send packet. In this module our aim is to send the packets safely to the destination without dropping the packets in between by avoiding Selfish & Malicious Nodes. So our overall logic is to eliminate selfish &

malicious path and best path is identified through the monitoring watchdog activity of available paths.

IV. CONCULSION

This paper proposes CoCoWa as a collaborative contact-based watchdog to reduce the time and improve the effectiveness of detecting selfish nodes, reducing the harmful effect of false positives, false negatives and malicious nodes. CoCoWa is based on the diffusion of the known positive and negative detections. When a contact occurs between two collaborative nodes, the diffusion module transmits and processes the positive (and negative) detections. Analytical and experimental results show that CoCoWa can reduce the overall detection time with respect to the original detection time when no collaboration scheme is used, with a reduced overhead (message cost). This reduction is very significant, ranging from 20 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 CoCoWa we can reduce the effect of malicious or collusive nodes. If malicious nodes spread false negatives or false, positives in the network CoCoWa is able to reduce the effect of these malicious nodes quickly and effectively. Additionally, we have shown that CoCoWa is also effective in opportunistic networks and DTNs, where contacts are sporadic and have short durations, and where the effectiveness of using only local watchdogs can be very limited.

REFERENCES

[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] [4] L. Butty_an and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in Proc. 1st Annu. Workshop Mobile Ad Hoc Netw. Comput., 2000, pp. 87–96.
[5] [5] L. Butty_an and J.-P. Hubaux, "Stimulating cooperation in selforganizing mobile ad hoc networks," Mobile Netw. Appl., vol. 8, pp. 579–592, 2003.
[6] [6] H. Cai and D. Y. Eun, "Crossing over the bounded domain: From exponential to power-law intermeeting time in mobile ad hoc networks," IEEE/ACM Trans. Netw., vol. 17, no. 5, pp. 1578–1591, Oct. 2009.
[7] [7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," IEEE Trans. Mobile Comput., vol. 6, no. 6, pp. 606–620, Jun. 2007.
[8] [8] J. R. Douceur, "The sybil attack," in Proc. Revised Papers 1st Int. Workshop Peer-to-Peer Syst., 2002, pp. 251–260.
[9] [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] [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] [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] [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] [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. 1174 IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 14, NO. 6, JUNE 2015
[14] [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] [15] J. Hortelano, J.-C. Cano, C. T. Calafate, M. de Leoni, P. Manzoni, and M. Mecella, "Black hole attacks in p2p mobile networks discovered through Bayesian filters," in Proc. Int. Conf. Move Meaningful Internet Syst., 2010, pp. 543–552.
[16] [16] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the usefulness of watchdogs for intrusion detection in VANETs," in Proc. Int. Conf. Commun. Workshop, 2010, pp. 1–5.
[17] [17] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., 2008, pp. 241–250.
[18] [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.