ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Real Time Selfish Behaviour Detection and Defence in Ad Hoc Networks

Reshma Kalesh

M.Tech Student (Communication Engineering), Dept of ECE, IIET, M.G. University, Kottayam, Kerala, India

ABSTRACT: In ad hoc networks, selfish nodes deviating from the standard MAC (Medium Access Control) protocol can significantly degrade normal nodes' performance and are usually difficult to detect. Then propose a detection and defence schemes to identify and defend against MAC-layer selfish misbehaviour, respectively, in IEEE 802.11 multi-hop ad hoc networks. According to the existing detection rule, a number of well behaved nodes may classified as selfish nodes. And the number of nodes that detected and isolated as selfish nodes are large. Therefore for the precise detection of selfish nodes some methods are used. Here uses a COCOWA (Collaborative Contact Based Watchdog) scheme to reduce the time and improve effectiveness of detecting selfish nodes. An efficient timer based acknowledgement technique is also proposed to detect and isolate the misbehaving nodes and can even find a possible alternate route in case when the number of misbehaving nodes is greater than minimum count.

KEYWORDS: MAC layer, Selfish misbehavior of nodes, IEEE 802.11 multi hop ad hoc network, Alternate route, Detection and isolation of selfish nodes.

I. INTRODUCTION

MAC layer misbehaviour can be generally classified into the following two categories: malicious misbehaviour and selfish misbehaviour . selfish users can deliberately deviate from the standard MAC protocol to gain more network resources over well behaved nodes. Generally, selfish nodes can benefit in the following two scenarios: first, obtaining a large portion of channel sharing, and second, reducing power consumption, e.g., by denying the forwarding of incoming packets. Compared to the second scenario, the first one is more difficult to detect, and can result in more serious problems and greatly degrade normal users' performances. In this paper, focus on the MAC layer selfish misbehaviour problem in wireless ad hoc networks, in particular, IEEE 802.11 ad hoc networks, where selfish nodes aim to obtain higher MAC layer performance .Here address this issue from two perspectives: detection and defence.

Selfish misbehaviour detection: In IEEE 802.11 networks, selfish nodes can manipulate the following MAC layer parameters to enhance their channel access probabilities: the remaining transmission duration contained in frames, SIFS duration, DIFS duration, and backoff time. The most challenging detection task is to detect backoff time manipulation . The difficulty primarily stems from the non-deterministic nature of IEEE 802.11 MAC that does not allow a straight forward way of distinguishing between a normal transmitter, which happens to select short backoff time, and a selfish node that intentionally selects short backoff time. All the traditional methods for the detection of selfish nodes are designed for wireless local area networks (WLANs) only and cannot be directly applied to multi-hop ad hoc networks. The main reasons are as follows. First, some previous schemes rely on a large amount of historical data to perform statistical detection. Second, many schemes like are based on throughput or delay models which are only valid in WLANs. Noticing the above problems, in this paper propose a real time detection scheme for multi-hop ad hoc networks, which requires only several samples and no prior knowledge of selfish nodes.

Moreover, since one selfish node is usually watched by multiple observers, they send their reports to a local cluster head who employs the majority rule to make the final decision. To complete this work, here also briefly discuss how to detect the manipulation of the other three MAC layer parameters as well. In addition, the detection scheme is carried out periodically. Once a node is determined by a local cluster head as a selfish node, a penalty scheme is applied to punish the detected selfish node by decreasing its throughput. Specifically, all its one-hop neighbours will stop forwarding packets for it until they receive another decision notice indicating that this node is not selfish any more.

In the existing system, when a node is detected as a selfish node the local cluster head just isolate the node without any further investigation of the nature of the selfish misbehaviour. According to the existing detection rule, a number of

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

well behaved nodes may classified as selfish nodes. And the number of nodes that detected and isolated as selfish nodes are large. Therefore for the precise detection of selfish nodes some methods are used. One of the method is, COCOWA, a collaborative contact based watchdog method to reduce the time and improve effectiveness of detecting selfish nodes. Here also uses an efficient timer based acknowledgement technique to detect and isolate the misbehaving nodes and can even find a possible alternate route in case when the number of misbehaving nodes is greater than minimum count. After the conduction of these tests only a node is finalised as a selfish node and isolated. Hence the number of nodes that detected and isolated as selfish nodes are very less.

II. RELATED WORK

There are a few works on the detection of backoff time manipulation. In [1], propose a sequential probability ratio test (SPRT) algorithm to address the detection problem. However, SPRT is a parametric statistical approach which means that prior knowledge of the selfish nodes' behaviour needs to be present at the detectors. Another system is DOMINO [2],it is non-parametric statistics hence they do not require exact models of the normal or adversarial distributions. They only require the knowledge of some of it's parameters. There are two extensions of DOMINO :they are O-DOMINO and CUSUM. These extensions still preserves the original intuition and simplicity of the algorithm, while significantly improving it's performance. In [3], develop a scheme in which a detected selfish transmitter would be required by the corresponding receiver to use a longer backoff time, assuming that the receiver is a normal node. A similar approach is proposed in another previous paper, designs game theoretic approaches for WLANs which are resilient to selfish misbehaviour. Besides, with modifications to the IEEE 802.11 binary exponential backoff (BEB) scheme, propose to force each node to generate a predictable contention window (CW) size. Any node who picks a smaller backoff value (contention window size) than the predicted one would be regarded as a selfish node.

In [4], in order to mitigate the misbehaviour of selfish nodes in degrading the network performance, an adaptive and predictable algorithm (PRB) is introduced. This is based on minor modifications of the IEEE 802.11 BEB. In PRB, like BEB the backoff value is selected randomly from the CW. And unlike BEB, lower bound of the CW for next transmission is predictable based on the current selected CW. PRB forces each node to generate a predictable backoff interval. Both PRB and BEB has the same result under normal case but when consider selfish misbehaviour case the result varied. The distributed nature of CSMA/CD based wireless protocols [5] allows malicious nodes to deliberately manipulate their back off parameters and thus unfairly gain a large share of the network throughput. Selfish misbehaviour involves disobeying standard protocol mechanisms to gain unfair access to the channel at the expense of other users. In [6] outline conditions on genuine (non-misbehaviour) node's throughput to guarantee the presence of misbehaviour, and propose non-adaptive and strong reaction mechanism for such aggressive misbehaviours. Both mechanisms are distributed in nature, rely only upon local information available at genuine nodes, and are thus easily implementable in practice. Proposed reaction mechanisms provide the necessary disincentive towards selfish misbehaviour, and are aimed at preventing misbehaviour.

In [7], proposes a secure MAC protocol for MAC layer which has integrated with a novel misbehaviour detection and avoidance mechanism for Mobile Ad Hoc Networks (MANETs). Common neighbours of the sender and receiver contributes effectively to misbehaviours detection and avoidance process at MAC layer. In [8], main focus on the impact of rushing attack implemented by malicious nodes (MNs) on AODV routing protocol as an extension . The malicious node that disobey the standard, degrades the performance of the well-behaved nodes drastically, even the entire network may collapse. In [9], propose an algorithm that ensures honest behaviour of non-colluding participants. Furthermore, analyse the problem of colluding selfish nodes, casting the problem within a mini max robust detection framework and providing an optimal detection rule for the worst-case attack scenarios. In [10], discusses two algorithms that are based on reputation based technique and one algorithm based on credit based technique. In Reputation based approach , in addition to punishing the selfish nodes, and encouraging the cooperating nodes, there is second chance for the nodes which dropped a packet unwillingly. In this approach if the node is recognized to be a selfish node for the first time and punished, the cooperation coefficient of it can be increased if it changes its behaviour as a co operator node.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

III. SELFISH MISBEHAVIOUR DETECTION

In this section, propose to detect selfish misbehaviour through normal node's observations. Specifically, here consider a multi-hop wireless network working on a single channel, in which every node is watched by all its neighbours. Normal nodes will compare the observed data with their counterparts under normal protocol operations, and apply the detection rules to determine whether the node under observation is a selfish node or not. Recall that in IEEE 802.11 networks, selfish nodes can manipulate four MAC layer parameters to gain higher channel access probability : the remaining transmission duration, SIFS duration, DIFS duration, and backoff time.

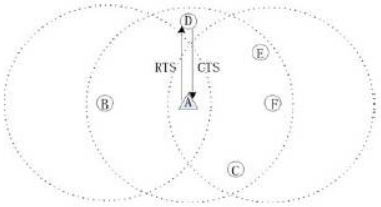

Fig.1. Illustration of network topology

(a) Detection of Remaining transmission duration Manipulation

Fig.1. shows a scenario where node A is the node of interest while nodes B to F are all A's one-hop neighbours, i.e, observers. In the remaining transmission duration manipulation case, node A sets the duration contained in the RTS/DATA frames it sends out to a larger value so that it can claim to occupy the channel for a longer period. Denote the duration contained in an RTS frame and the following DATA frame by DRTS and DDATA, respectively. When overhearing a DATA frame, nodes B to F can determine if A is a selfish node by checking the following two quantities:

$$\mathbf{T}R = \mathbf{D}RTS - 3\times\mathbf{SIFS} - \mathbf{T}CTS - \mathbf{T}DATA - \mathbf{T}ACK \quad ...(1)$$

$$\mathbf{T}D = \mathbf{D}DATA - \mathbf{SIFS} - \mathbf{T}ACK \quad ...(2)$$

where **T**CTS, **T**DATA and **T**ACK are the duration of CTS, DATA, and ACK frames respectively. **T**CTS and **T**ACK can be easily calculated based on the frame lengths specified in the IEEE 802.11 standard and the transmission rate, while **T**DATA can be measured by the observing nodes. If **T**R > 0 or **T**D > 0, i.e., the duration contained in RTS or DATA frames is larger than the real duration of the rest of the transmission, then A is a selfish node.

(b) Detection of SIFS manipulation

In this case, selfish nodes choose a smaller SIFS duration after receiving a CTS so as to finish its current transmission sooner. Then, nodes B to F can infer A's SIFS duration as follows:

$$\mathbf{T}SIFS = \mathbf{t}data - \mathbf{t}RTS - \mathbf{T}RTS - \mathbf{SIFS} - \mathbf{T}CTS \quad ...(3)$$

where **t**RTS and **t**DATA are the time instances at which an observer starts overhearing an RTS and the following DATA from A, respectively, and **T**RTS is the duration of RTS frames. If **T**SIFS < **SIFS**, then node A is a selfish node.

(c) Detection of DIFS duration/ Backoff time manipulation

Both DIFS duration manipulation and backoff time manipulation intend to shorten the waiting period to initiate transmissions after the channel is sensed idle. Reducing the DIFS duration is equivalent to choosing a smaller backoff time. Therefore, can detect these two cases using the same detection rules assuming the backoff time manipulation scenario. Since the backoff time is inherently a random variable, backoff time manipulation is much more difficult to detect compared to the previous two cases. Next, propose a detection scheme to detect the selfish nodes manipulating their DIFS duration or backoff time to access the channel with higher priority. Here mainly rely on the one-hop neighbours to perform the detection. Note that a normal node randomly chooses its backoff time, denoted by BT, from a contention window $[0, CW-1]$, where CW is its current contention window size.

Thus, the probability that a node's BT is less than or equal to an estimated backoff time $\boldsymbol{t}$ is :

$$P[BT \leq t] = \frac{t+1}{CW} \quad ...(4)$$

Where, CW can be inferred according to the discussions above. Besides, the above probability itself can also be considered a random variable, which we denote by X, since $\boldsymbol{t}$ is essentially uniformly chosen in the contention window. Thus, if a node is well-behaved, the expectation of X should be

$$E[X] = \frac{CW+1}{2CW} \quad ...(5)$$

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Since X is a random variable, it is very difficult to determine whether a node is a selfish node based on one single sample of X. Therefore, extend the above expression to the case of multiple observed samples in the following. Note that the backoff times of consecutive observed samples are correlated if they are for the same DATA frame (one DATA frame might be retransmitted several times with multiple backoff processes). Thus, an observer chooses multiple observation samples for different DATA frames which are thus uncorrelated and independent. In particular, consider *n* such chosen observation samples. Denote the detected backoff time and the corresponding contention window size for the *i*th ($1 \leq i \leq n$) sample by ***ti*** and *CWi*, respectively and

$$Xi = P[BTi \leq ti] = \frac{ti+1}{CWi} \quad ...(6)$$

Then, X_1,.....,Xn are independent of each other. Then can have their joint cumulative distribution function (CDF), which we denote by Y , as

$$Y = P[BT_1 \leq t_1,.....,BTn \leq t_n] \quad ...(7)$$

And the expectation as,

$$E[Y] = \prod_{i=1}^{n} \frac{CWi+1}{2CWi} \quad ...(8)$$

Thus, if an observer detects that,

$$Y \leq \mu E[Y], \quad ...(9)$$

it then considers the node under observation as a selfish node, where μ ($0 < \mu \leq 1$) is called the detection factor, a control parameter. In the detection rule, different values of μ will lead to different detection probabilities. How to set μ depends on the design goal want to achieve: if want to identify more selfish nodes in the networks, a larger μ is needed; on the other hand, if allow a few selfish nodes to exist and do not want the well-behaved nodes to be wrongly classified as selfish nodes, a smaller μ is more appropriate.

For instance, given a lower bound on the confidence level, then have an upper bound on the detection factor μ. Besides, usually one selfish node is watched by several observers. Each observer will employ the above detection scheme to perform detection and send their decisions to a local cluster head, which then employs the majority rule to make the final decision. Note that the proposed scheme needs only several data samples and no prior knowledge of selfish nodes. Thus, it is more efficient and takes less time than previous schemes, which rely on a large amount of historical data to perform statistical detection. Moreover, the proposed detection is carried out periodically. Once a node is determined by a local cluster head as a selfish node, a penalty scheme is applied to punish the detected selfish node by decreasing its throughput for a certain time period. Specifically, all its one hop neighbours will stop forwarding packets for it until they receive another decision notice indicating that this node is not selfish any more. In other words, each detection result is effective until a new detection result is obtained. Moreover, scheme discussed above deals with a single selfish node and does not include the scenario of colluding nodes. Recall that observers will send their detection results to a local cluster head, which then employs the majority rule to make the final decision. Therefore, as long as there are enough honest nodes (or more honest nodes than selfish nodes), selfish nodes can still be detected even though they collude with each other.

IV. PROPOSED METHOD

In the existing system, when a node is detected as a selfish node the local cluster head just isolate the node without any further investigation of the nature of the selfish misbehaviour. According to the existing detection rule, a number of well behaved nodes may classified as selfish nodes. Therefore for the precise detection of selfish nodes some methods are used. One of the method is, COCOWA, a collaborative contact based watchdog method to reduce the time and improve effectiveness of detecting selfish nodes. Here also uses an efficient timer based acknowledgement technique to detect and isolate the misbehaving nodes and can even find a possible alternate route in case when the number of misbehaving nodes is greater than minimum count. After the conduction of these tests only a node is finalised as a selfish node and isolated. Hence the number of nodes that detected and isolated as selfish nodes are very less.

(1) Detecting the MAC selfish node using Collaborative Contact based Watch dog Scheme

Here introduces Collaborative Contact-based Watchdog (COCOWA) 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,

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

nodes have second hand information about the selfish nodes in the network. The goal of this approach is to reduce the detection time and to improve the precision by reducing the effect of both false negatives and false positives.

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 retransmitted. 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 in Fig.2.It is based on the combination of a local watchdog and the diffusion of information when contacts between pairs of nodes occurs. A contact is defined as an opportunity of transmission between a pair of nodes (that is, two nodes have enough time to communicate between them). Assuming that there is only one selfish node, the figure shows how initially no node has information about the selfish node. When a node detects a selfish node using its watchdog, it is marked as a positive, and if it is detected as a non selfish node, it is marked as a negative. Later on, when this node contacts another node, it can transmit this information to it; so, from that moment on, both nodes store information about this positive (or negative) detections. Therefore, a node can become aware about selfish nodes directly (using its watchdog) or indirectly, through the collaborative transmission of information that is provided by other nodes. Under this scheme, the uncontrolled diffusion of positive and negative detections can produce the fast diffusion of wrong information, and therefore, a poor network performance.

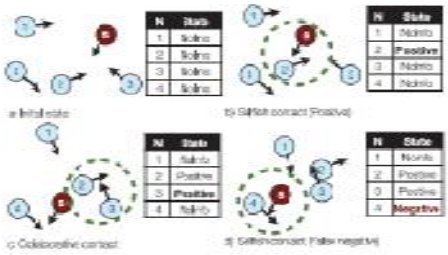

Fig .2. Working Principle

(2) A Timer based Acknowledgement Scheme for node misbehaviour detection and isolation

An efficient timer based acknowledgement technique is proposed to detect and isolate the misbehaving nodes and can even find a possible alternate route in case when the number of misbehaving nodes is greater than minimum count. This involves a detection timer and forward counter that help to reduce the number of acknowledgements thus reducing the delay and overhead. This approach is keenly focusing on acknowledgement of nodes regarding the misbehaviours so that the source takes the corresponding action. The advantage in this approach is that there is no need of sending acknowledgement for reception of each data packet since it is processed in group-wise and it minimizes the waiting period for acknowledgement and also overhead is reduced. Misbehaving nodes do not forward data packets and also they do not drop acknowledgement packets. False acknowledgement packets are never sent or forwarded by misbehaving nodes. In this scheme, it is assumed that, each node maintains a LIST which contains ID of every data packets sent or forwarded.

(a) Grouping of nodes and transmission of acknowledgement packets

As soon as the desired route is found, all the nodes of the desired route are logically grouped into N sets (i.e. M1, M2, M3…Mn), where M=m/3 (m is the number of nodes in the desired route) such that the group M1 contains first three consecutive nodes and group M2 contains next three consecutive nodes (as in Figure 1) and so on. Hence a group M1 consists of the source S which is First node referred as FNode and the intermediate node referred as I Node and the Last node of the group is referred as L Node.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

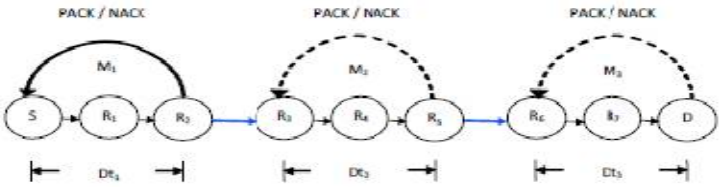

Fig.3. Grouping of nodes and transmission of PACK / NACK

For example if S→R1→R2→R3→R4→R5→R6 →R7→D is the desired route then nodes of the desired path forms three groups (i.e, M1, M2, M3).

The Group M1=S→R1→R2
Group M2=R3→R4→R5
Group M3=R6→R7→D

The proposed work focuses on detecting selfish nodes which drop packets such that the other nodes can never use it. Here the selfish behaviour of the nodes is considered as misbehaving because they drop packets to save battery power.

In order to track the incoming packets and outgoing packets a forward counter Fc is used in each node. The forward counter is updated when a packet leaves the node and when a packet enters the node. A detection timer D_{timer} is assigned for every group of nodes with specific time interval. When the Dtimer starts the source node, i.e F Node starts forwarding the packets and when the Dtimer expires, the last node say L Node send as acknowledgement to the S Node. The proposed scheme aims to detect and isolate the misbehaving nodes.

(b) Detecting misbehaving nodes

The nodes start forwarding the packet upon request. When this action begins, the D timer starts. A forward counter is used to update the packets entering and leaving the node. Then the forward counter is on and this gets incremented or decremented according to the flow. When the packet enters the node, the Fc is incremented and when the packet leaves node Fc is incremented. After the Dtimer expires, the last hop node of the group compares the value of Fc with forward counter threshold Fct. If Fc of L Node is equal to Fct then the source is informed with the positive acknowledgement , PACK otherwise with negative acknowledgement NACK. In this manner the process continues for every group of nodes . The merit of this approach is that there is no need of sending acknowledgement for reception of each data packet since it is processed in group-wise and it minimizes the waiting period for acknowledgement and also overhead is reduced.

(c) Mitigating misbehaving nodes

If the source is informed with PACK, the route is considered as normal. If NACK is informed to the source node, then the source node of every group counts the NACK of each node. If NACKc is greater than NACKcmax, then the node is considered as misbehaving and this information is broadcasted to all other groups in the route. From the broadcast information, the destination node checks the number of misbehaving nodes along the route and this information is sent as a feedback to the source node. If the source node finds that only limited number of misbehaving nodes in the route, then that particular nodes are marked as rejected and bypass route is established excluding those nodes. When the number of misbehaving nodes exceeds the minimum count, then the entire route is treated as misbehaving and an alternate route is established for the transmission, by the source. This process is efficient technique for route discovery since the delay and overhead is reduced.

V. SIMULATION RESULTS

(A) Selfish Misbehaviour Detection

Simulations are performed in NS2 to demonstrate and evaluate the performance of the proposed method. Here, carry out simulations in a network of 20 nodes to verify the performance of selfish misbehaviour detection schemes. Since the detection of remaining transmission duration manipulation and of SIFS manipulation are comparatively easier, here mainly focus on the detection of DIFS duration/ backoff time manipulations. Simulation results of average channel occupation duration when ρ=10,30,50 percent are given in the graphs below. Simulation results when selfish nodes are naive, random, and γ-persistent are shown in Figs,4,5,6, respectively.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

In Fig.4. as the constant backoff value adopted by naive selfish nodes grows, selfish node's average channel occupation duration drops while normal node's average channel occupation duration slightly increases. In fig.5. as the conduction window size adopted by random selfish nodes increases, selfish node's average channel occupation duration drops while normal node's average channel occupation duration slightly increases. In fig.6. as the value of the control parameter γ adopted by γ-persistent node grows, selfish node's average channel occupation duration drops while normal node's average channel occupation duration slightly increases.

Where selfish nodes and normal nodes have the same channel occupation duration when CW=45 and γ=0.96, respectively. Besides, denote the percentage of selfish nodes among all the nodes as ρ, call "selfish node density". When ρ = 50%, i.e., when 20 × 50% = 10 nodes are randomly selected as selfish nodes, the channel occupation durations of selfish nodes and of normal nodes are lower than those of selfish nodes and of normal nodes, respectively, when ρ = 30% and when ρ = 10%. The reason is that the selfish nodes will compete with each other, besides with normal nodes, to access the channel. The more selfish nodes there are, the lower their average channel occupation duration would be.

(B) Selfish Misbehaviour Punishment

Here analyse the impact of selfish misbehaviour penalty scheme on both selfish nodes and normal nodes. Here consider a network of 20 nodes with selfish node density equal to 30 percent. Besides, here also consider both dumb selfish nodes and smart selfish nodes. Once a node is determined as a selfish node, all its one-hop neighbours will stop forwarding packets for it until a different decision is made. In this simulations, let cluster heads collect detection results from observers and feed back the decision results to the observers every 5 seconds.

Fig.7. shows the throughput of a randomly selected smart selfish node, the throughput of one of its one-hop normal nodes, and that of one of the other normal nodes. From this can observe that the throughput of the three nodes are nearly the same most of the time. Notice that the throughput of the smart selfish node has two sharp drops. Then can infer that the selfish node is detected as being selfish twice during the simulation period. The reason why the throughput does not decrease to 0 is that some other nodes in the network have buffered some packets for the selfish node and can still forward these packets to the destinations. In addition, Fig.8. shows the throughput of a randomly chosen naive selfish node. Here find that the naive selfish node's throughput has two sharp drops, indicating that it has been judged as a selfish node twice during the entire simulation time.

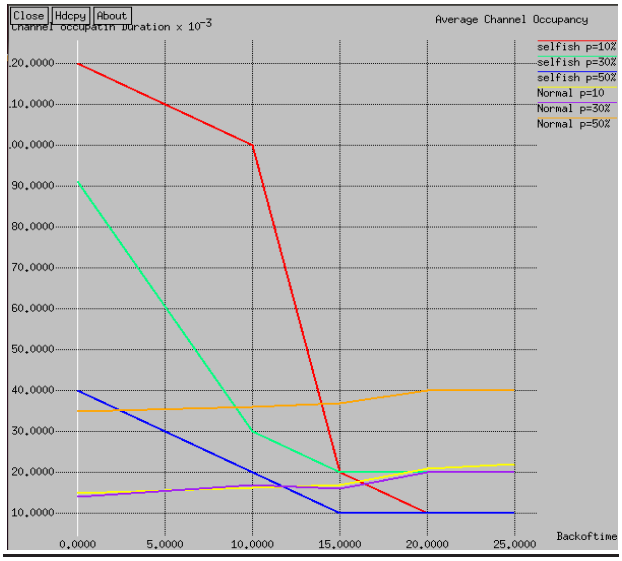

Fig.4.Average channel occupancy of Naive selfish nodes.

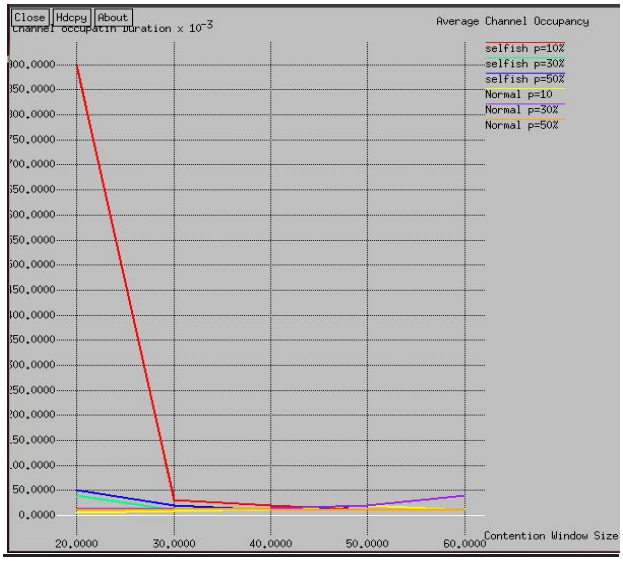

Fig.5.Average channel occupancy of Random selfish nodes

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

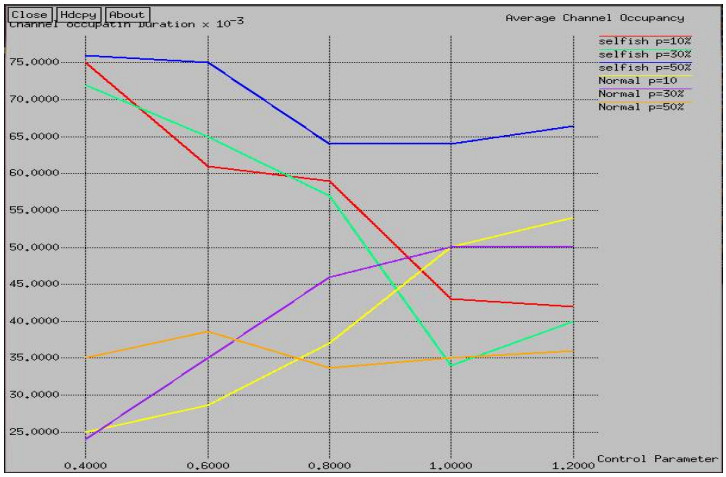

Fig.6.Average Channel Occupancy of γ-persistent selfish nodes

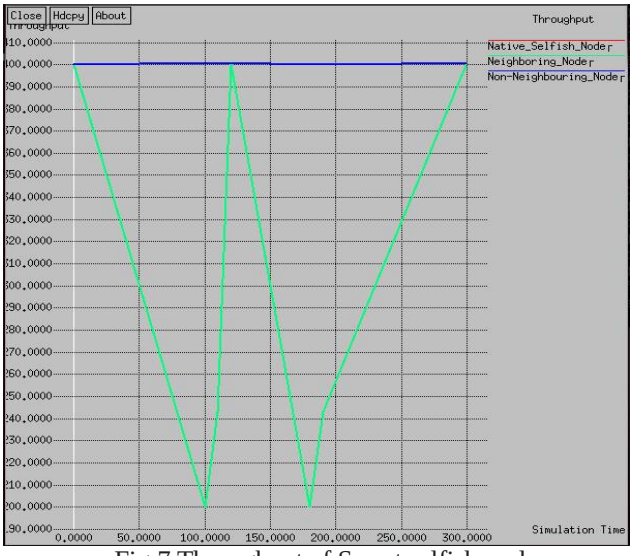

Fig.7.Throughput of Smart selfish nodes

Fig.8.Throughput of Naive selfish node

VI. CONCLUSION

In wireless ad hoc networks, selfish nodes that deliberately deviate from the standard MAC protocol may obtain an unfair share of the channel resource and degrade the performance of other well-behaved nodes. This type of misbehaviours are usually difficult to detect. Most traditional selfish misbehaviour detection approaches are rely on a large amount of historical data to perform statistical detection that are only valid in WLANs for detection. Here presented a distributed observation based selfish misbehaviour detection scheme for multi hop ad hoc networks. It requires only several samples. In the existing system , the number of nodes that detected as selfish nodes are greater. Which may leads to the generation of void regions in the network. Then between the source and destination path Bypassing is required for the transmission of packets. In the proposed system, for the precise detection of selfish nodes, some tests are conducted. Here Collaborative Contact Based Watchdog scheme is used to reduce the time and improve effectiveness of detecting selfish nodes. An efficient timer based acknowledgement technique is also proposed to detect and isolate the misbehaving nodes and can even find a possible alternate route in case when the number of misbehaving nodes is greater than minimum count. Hence compared to the existing system the number of nodes that detected as

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

selfish are very less. Therefore the number of nodes isolated will be less. The advantages of the proposed methods are , the chances to interpret the normal node as a selfish node can be avoided.

REFERENCES

1. S. Radosavac, J. S. Babaras, and L. Koutsopoulos, "A framework for MAC protocol misbehaviour detection in wireless networks," in Proc. 4th ACM Workshop Wireless Security, Cologne, Germany, pp. 33–42 , Sep. 2005.
2. M. Raya, J. Hubaux, and I. Aad, "Domino: A system to detect greedy behaviour in IEEE 802.11 hotspots," in Proc. ACM 2nd Int. Conf. Mobile Syst., Appl. Serv., Boston, MA, USA , pp. 84–97, Jun. 2004.
3. P. Kyasanur and N. Vaidya, "Selfish MAC layer misbehaviour in wireless network," IEEE Trans. Mobile Comput., vol. 4, issue 5, pp. 502–516, Sep. 2005.
4. L. Guang, C. M. Assi, and A. Benslimane, "Enhancing IEEE 802.1 random backoff in selfish environment," IEEE Trans. Veh. Technol., vol. 57, issue 3, pp. 1806–1822, May 2008.
5. J. Tang, Y. Cheng, and W. Zhuang, "An analytical approach to real-time misbehaviour detection in IEEE 802.11 based wireless networks," in Proc. IEEE Conf. Comput. Commun., Shanghai, China,, pp. 1638–1646, Apr. 2011.
6. N. Jaggi, V. R. Giri, and V. Namboodiri, "Distributed reaction mechanisms to prevent selfish misbehaviour in wireless ad hoc networks," in Proc. IEEE Global Telecommun. Conf., Houston, TX, USA, pp. 1–6, Dec. 2011.
7. Guang, L., Assi, C., Benslimane, A. "MAC layer Misbehaviour in Wireless Networks: Challenges and Solutions" In Journal of Computer Security, vol. 14, issue 4 , pp. 6-14, 2008.
8. S. Choudhury, N. Chaki "Routing Misbehaviour in Ad Hoc Network", Int. Journal of Computer Appl. vol 1,issue 18, pp 0975-8887, 2010.
9. S. Radosavac , Alvaro A. Cárdenas , John S. Baras and George V. Moustakides "Detecting IEEE 802.11 MAC layer misbehaviour in ad hoc networks: Robust strategies against individual and colluding attackers", Journal of Computer Security.vol 15,pp.103-128,2007.
10. Dipali Koshti, Supriya Kamoji "Comparative study of Techniques used for Detection of Selfish Nodes in Mobile Ad hoc Networks",Int. Journal of Soft Computing and Engg., vol 1,issue 4, pp 2231-2307, Sep. 2011.