MODSPIRITE: A Credit Based Solution to Enforce Node Cooperation in an Ad-hoc Network

Rekha Kaushik[1], Jyoti Singhai[2]

[1] Department of Information Technology, MANIT
Bhopal, Madhya Pradesh, India

[2] Department Of ECE, MANIT
Bhopal, Madhya Pradesh, India

Abstract

In an Ad-hoc network, node cooperation is an important factor for successful data transmission among nodes. Most of the routing protocols are based on the assumption that the intermediate node cooperates during data transmission from source node to destination. However, because mobile nodes are constrained by limited energy, bandwidth and computational resources, a node sometimes behaves as selfish to conserve its resources like energy, bandwidth etc. These selfish nodes are unwilling to forward others' packets. This paper gives a review of existing reputation based and credit based systems and proposes a credit based solution called MODSPIRIT which is a modification of SPIRITE system. MODSPIRITE system detects selfish node using neighbor monitoring mechanism and enforce cooperation among non cooperative node by providing incentives to intermediate nodes. One of the limitations of SPIRITE system is that sender loses too much credit to forward its data to the destination and for future sender have very less or no credit to forward its data. As compared to SPIRITE, MODSPIRITE reduces overhead of sender for upto 25%. It also punishes non-cooperative nodes so that non cooperative node get discourage.

Keywords: *Ad-hoc Network, Node Cooperation, Selfish node, Credit based system.*

1. Introduction

An Ad-hoc network (MANET) is a self configuring and infrastructure less network of mobile nodes. Each node acts as a router and free to move independently in any direction. In an ad-hoc network communication between two nodes beyond the transmission range relies on intermediate nodes to forward the packet. The communication takes place using routing protocol [1] which is of three types: Proactive, Reactive and Hybrid routing protocol.

Pro-active (table-driven) routing: This type of protocols such as DSDV maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. The main disadvantage of such algorithms is slow reaction on restructuring and failures.

Reactive (on-demand) routing: This type of protocols such as DSR, AODV[2] finds a route on demand by flooding the network with Route Request packets. The main disadvantages of such algorithms are high latency time in route finding and excessive flooding can lead to network clogging.

Hybrid routing: This type of protocols combines the advantages of proactive and of reactive routing. The routing is initially established with some pro actively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. The choice for one or the other method requires predetermination for typical cases. The main disadvantages of such algorithms are advantage depends on amount of nodes activated and reaction to traffic demand depends on gradient of traffic volume.

This paper uses DSR [2] which is a source routing protocol and this protocol can react to topological changes rapidly. DSR is a reactive routing protocol. There are two main operations in DSR; route discovery and route maintenance. Each node gathers information about the network topology by overhearing other nodes' transmissions. This is known as promiscuous mode of operation. Each node maintains a route cache to remember routes that it has learnt about. All of the routing protocols including DSR assume that all nodes in the network are cooperative and forward others' messages.

However, since each node in an Ad-hoc network is constrained by limited energy, bandwidth and computational resources, a node may not be willing to forward packets that are not directly beneficial to it or node attack on routing protocol to disrupt network performance. Such nodes are known as Non-cooperative or misbehaving

nodes, which can be classified as selfish nodes [3] and malicious nodes [3].

A selfish node does not forward any data packets for other nodes except for itself to conserve its resources (energy, bandwidth) while Malicious node injects false information and/or removes packets from the network to sabotage other nodes or even the whole network.

Numerous methods have been proposed to deal with the problem of selfish and malicious nodes. These methods can be divided into two categories: Reputation based system [4][5][6][7][8][9] and credit (Incentive) based system[10][11][12]. Reputation based system watches others' behaviour to detect misbehaving nodes. If a node is selfish and drops other nodes' packet, it will earn bad reputation and will be isolated by other nodes. Credit based system rewards a node with certain credit when it forwards the packet of other node. If a selfish node does not forward other nodes' packet, it loses credit and ultimately it is left with insufficient or no credit at all to forward its own data.

One of the earlier work based on credit based system is Sprite system [5]. SPRITE (simple, cheat- proof, credit-based system) for mobile ad-hoc networks with selfish nodes, uses credit to provide incentive to cooperative nodes. When a node receives a message, it keeps a receipt of the message. Later, when the node has a fast connection to a Credit Clearance Service (CCS), it reports to the CCS the messages have been received/forwarded by uploading its receipts. The CCS then determines the charge and credit to each node involved in the transmission of a message, depending on the reported receipts of a message. There are some limitations of SPRITE system; firstly, there is an excessive burden on sender which loses credit for forwarding of its message. Secondly no punishment scheme is there for selfish nodes and also there is ambiguity between the nodes as to which one is selfish node.

This paper proposes a credit based solution called MODSPIRIT to enforce cooperation among non cooperative nodes. This system is modification of SPIRITE system.

The basic scheme of proposed algorithm is that when a node receives a message, it keeps a receipt of the message. It then communicates with the cluster head which is responsible for credit and debit of charges to nodes when they receive/forward messages to other nodes. Usage of cluster head reduces the burden of tamper proof hardware or CCS. Detection of selfish node is carried out by using neighbor monitoring mechanism as discussed in section 4. This mechanism is applied on limited number of intermediate nodes; hence reduces the computing overhead as described in earlier reputation based system.

On comparing the SPRITE system and the MODSPRITE system, the MODSPRITE system reduces burden on sender which loses credit for forwarding its message. As number of nodes increases in the network, the sender overhead reduces gradually. Punishment on selfish node given by sender encourages nodes to cooperate. Using cluster head instead of CCS reduces the burden of extra hardware and software. It reduces single points of failure. If CCS fails, the overall credit scheme fails while if cluster head fails, operations can simply transfer to other node.

The remainder of this paper is organized as follows. Related work is discussed in Section 2, which includes a description of Reputation based mechanism, Credit based mechanism and Credit cum Reputation based mechanism. Section 3 describes overview of SPIRITE protocol. Overview of MODSPIRITE is discussed in section 4. Proposed modification is presented in section 5 followed by Conclusion and Future work in Section 6.

2. Related Work

Since enforcing node cooperation for transferring other nodes' packets is a major concern in an ad-hoc network. Most of the existing solutions are based on following mechanisms: reputation based, credit based system and Reputation cum Credit based System.

2.1 Reputation based mechanism.

In an Ad hoc network, Reputation systems are used to keep track of the quality of behaviour of other node. Basically reputation is an opinion formed on the basis of watching node behaviour. Reputation can be calculated by direct observation and/or indirect observation of the nodes, through route or path behaviour, number of retransmission generated by the node , through acknowledgement message and by overhearing node's transmission by the neighbouring nodes [4][5][6][7].

One of the main goals/reasons for reputation systems to be used in a network of entities interacting with each other is to provide information to help assess whether an entity is trustworthy. This helps in detection of selfish and malicious nodes. Another goal is to encourage entities to behave in a trustworthy manner, i.e. to encourage good behavior and to discourage untrustworthy entities from participating during communication.
Reputation system exchange reputation values which have to be taken care off as malicious node can attack on such messages. Different kinds of attack are as follows:

- Spurious rating - Node could lie and give spurious rating information.

- Self-Promoting - Attackers manipulate their own reputation by falsely increasing it.
- Whitewashing - Attackers escape the consequence of abusing the system by using some system vulnerability to repair their reputation. Once they restore their reputation, the attackers can continue the malicious behavior.
- Slandering - Attackers manipulate the reputation of other nodes by reporting false data to lower the reputation of the victim nodes.
- Denial of Service - Attackers cause denial of service by preventing the calculation and dissemination of reputation values.
- False rumour: In false rumor misbehavior, a node floods the false and negative information regarding other nodes by claiming that they are misbehaving but actually they are not.
- Collusion: In this attack, two or more nodes collude in order to influence the reputation rating. Here a node can recommend others node as cooperative or can give negative information of cooperative node.

Watchdog/pathrater [4] is basic and most popular mechanism for the detection of misbehaving nodes. It uses the benefits of promiscuous mode of dynamic source routing protocol [2], in which a node can overhear the transmission or communication of its neighbours. Watchdog detects the misbehaving node by overhearing the communication and compares the message with the data stored in its buffer. If the data doesn't match, then after a threshold value the source of the concerned path is informed. Pathrater maintains rating of every used path. Nodes select routes with the highest average node rating.

This method suffers from various problems like: ambiguous collision problem, receiver collision problem, limited transmission power, collusion of nodes and partial dropping. Also, in this method the misbehaving node gets isolated, so this becomes reward for misbehaving node and its sole intention of energy saving is accomplished. This method can only detect the selfish node but unable to do anything to correct it.

CORE, a collaborative reputation mechanism proposed by Michiardi and Molva [5], has a *watchdog* component. however it is complemented by a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. CORE permits only positive second-hand information, which makes it vulnerable to spurious positive ratings and misbehaved nodes increasing each other's reputation.

Buchegger and Boudee proposed CONFIDANT[6] protocol which uses reputation mechanism to identify and isolate selfish nodes. The protocol is based on selective altruism and utilitarianism, thus making misbehavior unattractive. CONFIDANT consists of four important components - the Monitor, the Reputation System, the Path Manager, and the Trust Manager. They perform the vital functions of neighborhood watching, node rating, path rating, and sending and receiving alarm messages, respectively. Each node continuously monitors the behavior of its first-hop neighbors. If a suspicious event is detected, details of the event are passed to the Reputation System. Depending on how significant and how frequent the event is, the Reputation System modifies the rating of the suspected node. Once the rating of a node becomes intolerable, control is passed to the Path Manager, which accordingly controls the route cache. Warning messages are propagated to other nodes in the form of an *Alarm* message sent out by the Trust Manager.

Self policing MANET [7], combines misbehavior detection method with reputation system. Here each node can make their own decision on how to react to the behaviour of other node. Self policing provides a disincentive for cheating by excluding node from network. In this paper, author enhances CONFIDANT protocol and maintains two rating to make decision about the node: reputation rating and trust rating.

In [8], the mechanism relies on the principle that a node autonomously (without communicating with other neighbouring node) evaluates its neighbor based on the completion of request services. On successful delivery, reputation index increases else decreases. This can be done through TCP acknowledgement. It provide detection, prevention and punishment scheme to misbehaving nodes. In this paper, the author does not discuss about the value of reputation threshold chooses.

COSR [9](Cooperative On Demand Secure Routing Protocol) , is an extension of DSR protocol that uses reputation model to detect malicious and selfish behaviour of node and make all nodes more cooperative. In COSR Fei Wang measures node reputation and Route reputation using three parameters: *contribution of node* (how many route as well as data packet are forwarded between nodes), *capability of forwarding* packet of a certain node using energy and bandwidth threshold and *recommendation* which represent other's subjective recommendation. Advantage of COSR is that it is capable of avoiding hot points.

However, there are limitations of reputation based mechanism. First, as there is a possibility of collision, a packet will naturally drop even in the absence of a selfish node. This makes it difficult to ascertain whether the packet

drop is due to natural reasons or selfish behaviour of node. Second, the selfish nodes isolated from the network using reputation based scheme cannot be used in data forwarding. This solution is trivial, but not efficient. Much approach does not punish nodes that do not cooperate since data is forwarded using a different path without complaint. Another limitation of reputation based system is that they often assume that nodes that send reputation information about their peers are themselves trustworthy; and they are subject to collusion among nodes that misreport reputation information

2.2 Credit based mechanism.

Credit based system also known as incentive based system reward nodes for forwarding by giving those credits. Without credit, a node cannot transmit self-generated data packets.

Butty´an and Hubaux proposed incentives to cooperate by means of so-called nuglets [10] that serve as a per-hop payment in every packet in a secure module in each node to encourage forwarding. The secure module is required to ensure the correct number of nuglets is withdrawn or deposited. They propose two models for the payment of packet forwarding, the Packet Purse Model and the Packet Trade Model. In the Packet Purse Model the sender pays and thus loads the packet with a number of nuglets. Each intermediate node takes one nuglet when it forwards the packet. If there are no nuglets left at an intermediate node, the packet is dropped. If there are nuglets left in the packet once it reaches the destination, the nuglets are lost. In the Packet Trade Model, the destination pays for the packet. Each intermediate node buys a packet from the previous hop and sells it to the next for more nuglets. Since charging the destination and not the sender can lead to an overload of the network and the destination receiving packets it does not want, mainly the Packet Purse Model is considered. This model, however, can lead to the loss of nuglets which have to be re-introduced into the network by a central authority.

Zhong et al [11] propose an incentive based system named SPRITE, in which selfish nodes are encouraged to cooperate. In this system, a node reports to the Credit Clearance Service, the messages that it has received/forwarded by uploading its receipts. Intermediate node earns credit when they forward message of others' node. In addition to the availability of central authority, sprite assumes source routing, and a public key infrastructure.

In [12] describe a wireless health monitoring system using incentive based router cooperation. This system uses two cooperation protocol Continuous value (CVCP) cooperation protocol and discrete value cooperation protocol (DVCP) for improving message delivery reliability. CVCP protocol check the value of offered and stored credits. DVCP is designed for smaller ad hoc network and uses approximate values such as high (H), medium (M), low (L) to represent the incentive. This paper sets credits on the basis of priority of message.

Limitations of this mechanism are, a virtual bank is required to manage credits. Secondly, when a node has enough credits to send its own data, it can decide not to cooperate anymore and starts dropping packets. Routing overhead is high when credit based mechanism is used. Also, securing messages containing credits is also an essential requirement so that malicious node could not change credit value. They did not pay attention to the fairness issue in routing when some nodes do not get any reward due to some reason e.g.: location.

2.3 Reputation cum Credit based System

Secure and Objective Reputation–based Incentive (SORI) scheme [13] encourages packet forwarding and disciplines selfish behaviour in a non cooperative ad hoc network. Reputation of the node is used as an incentive to cooperate among nodes. Authors are able to design a punishment scheme to penalize selfish nodes.

ARM [14] selects low mobility nodes as reputation management nodes and is responsible for managing reputation values. ARM uses locality aware Distributed Hash Table for efficient reputation information collection and exchange. Advantage of using ARM is that ARM builds a hierarchical structure to efficiently manage the RVs of all nodes, and release the reputation management load from individual high mobility nodes. This enables low overhead and fast global reputation information accesses. Also ARM does not require currency circulated in the system.

From above literature survey, following issues will be considered to make comparison for different mechanism.

Detection of non-cooperative node: Both reputation based system and credit based system uses one of the following technique for the detection of non cooperative node. Promiscuous mode is used to overhear the communication of their neighboring node as in [4]. In core nodes do not only rely on promiscuous mode, but in addition they can judge the outcome of a request by rating end to end connection. In [6] monitor mechanism is used and neighbour watch mechanism is used by [11][13]. Retransmission of message, route reply message[8] and history or previous observation are also used by different authors to detect non cooperative nodes.

Management devices: Both reputation and credit based mechanism require devices or nodes for the management of reputation value or credit value. In SPIRITE[11], Credit

Clearance Service (CCS) is there for the credit management, ARM[13] uses low mobility devices for reputation management. Neighboring nodes are used to keep reputation as in [8][9]. Reliable clearance service is used in [14]. Various parameters are used to choose management nodes, for example high energy or battery unit, locality, reputation table and cost credit unit. Cluster head is used as a credit management node in this paper.

Robustness against non-cooperative node: Systems like CONFIDANT effectively prevent network from malicious node, also it motivate selfish node to cooperate. SORI, ARM , [8] work well with selfish nodes. COSR works well with blackhole, wormhole, rushing attack and selfish node but is unable to handle DOS attack.

Robustness against collusion: SPIRITE, CONFIDANT is collusion resistant system.

Global / Local Reputation or credit management: From above references it is carried out that reputation value is kept either globally or locally. Each has advantage as well as disadvantage. In global Reputations maintaining, each node maintains reputation values of every other node, so the size is O(N) while in Local Reputation each node maintains reputation values of the neighbor node that is located in one-hop. Global reputation/credit management needs an additional computational overhead. Global reputation has to decide whether to accept or reject a warning message and to update the reputation table. Local reputations are less vulnerable to false accusations than global reputations because it uses direct observation.

Global reputation/credit management are less reliable as message traverse across the network so that it could be delayed, modified, replayed or accidentally lost during the transmission. Global reputation has better performance with respect to the mobility issue, because every node knows the behaviour of other node in the network so possibility to cheat is less.

Authentication mechanism: Spirite uses cryptographic method and digital signature to prevent data from non cooperative node. The major increased overhead is the use of digital signature for message authentication. On comparing RSA with ECNR [11], ECNR uses much smaller bandwidth and storage requirement. The propagation of reputation is computationally and efficiently secured by a one-way-hash-chain-based authentication scheme [12]. One-way-hash function is computationally much cheaper than the digital signature. Author in [14][15] utilize hash chains to reduce number of digital signature operation.

3. Overview of Spirite

Figure 1 shows the overall architecture of SPIRITE system which consist of Credit clearance service (CCS) and a collection of mobile nodes. Nodes are equipped with network interface that allow node to send and receive message.

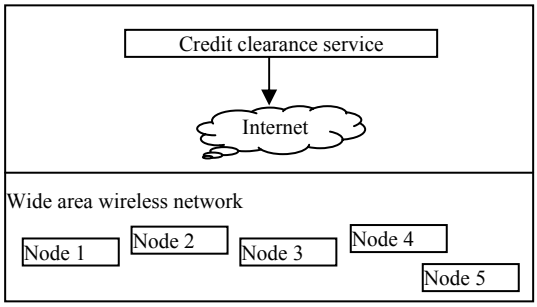

Figure 1: The architecture of SPIRITE

A node reports to the CCS, the messages that it has received/forwarded by uploading its receipts. Intermediate nodes earn credit when they forward message of others' node.

For motivating nodes to forward packet, the CCS determines the last node on the path that has ever received the message. Then CCS asks the sender to pay β to this node, and α to each of its successors. Here α is considered to be one and β is a very small value (for eg: 0.01). Figure 2 illustrate the payment system. According to the scenario as taken in figure 1, the sender pays a total of 2α+ β credits.

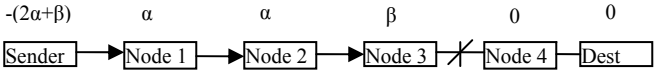

Figure 2: Illustration of the payment system

According to SPIRITE, charging sender will be more robust because of two reasons. Charging the destination may allow other nodes to launch DOS attack on the destination by sending large amount of traffic to it. On the other hand, if only the sender is charged, a node will not have incentives to send useless message.

To prevent from colluding nodes, payment scheme should be revised. For this CCS charges the sender an extra amount of credit if the destination does not report the receipt of the message. This extra charge goes to CCS instead of nodes.

Limitation of SPIRTE:

There are some limitations of SPIRITE system such as there is too much burden on sender. For example let us consider that there are n+1 node in the network and nth node act as selfish node then the burden on sender will be

(n-2)α+β ie, (n-2)α+β credit is lost by sender. If a sender wants to send huge data, then for the next time, the sender does not have enough credit to forward its own message.
In figure 2, there is dilemma whether it is node 3 or node 4 that drops the receipt/message. There is no scheme defined to solve this stage of ambiguity. Also there is no punishment for the node that does not forward the message. So this becomes reward for the non cooperative node.

The proposed modification called MODSPIRITE detects selfish node using neighbor monitoring mechanism and solves the stage of ambiguity. It also overcomes the above problems of SPIRITE by reducing overhead of sender by decreasing the incentive given by sender. It also gives punishment to non cooperative node to encourage cooperative nodes and discourage non cooperative nodes.

4. Modspirite System

The architecture of MODSPIRITE system contains several nodes and a cluster head. Consider there are m numbers of nodes in a network and the sender wants to send data to destination through intermediate node 1, node 2 and so on as shown in figure 3. A cluster head present in the network provides service to manage credit exchange mechanism in the network. All other nodes communicate with cluster head and give receipt of forwarding data packet. Cluster head selection criteria can base on ID, degree, residual energy, low mobility and association with other nodes.

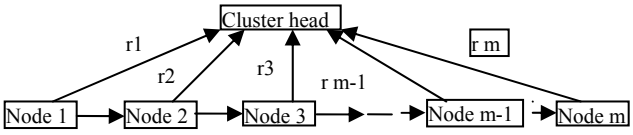

Figure 3: Architecture of MODSPIRITE

Initially each node has fixed amount of credit which is the essential requirement for the sender to forward its message. When a source node wants to send message to another node (destination), it will lose credit. The credit will be earned by the intermediate nodes which are responsible for forwarding the message. To earn more credit, a node must forward others' message.

Nodes report the forwarding of data packet to Cluster head in the form of small message called receipts which contain information like forwarding node address, destination address, and number of bytes sent. For example, node 1 send r1 receipt node 2 send r2 and so on as shown in figure 3. This paper assumes that cluster head is associated with all nodes in the network and always have sufficient amount of resources. Cluster head serves the purpose of managing credits. Nodes communicate with the cluster head after transferring their data. Only sender loses credit to forward its data. Credit is a virtual integer value. This paper considers only selfish node and not malicious nodes. Earning and losing of credit is applied only when there is presence of selfish node in the path.

5. Proposed modification

Consider all nodes initially have sufficient credit to facilitate forwarding of messages. If data correctly reaches the destination, no credit is lost or earned by the sender and intermediate nodes respectively. If the data does not reach the destination, it indicates that one of the intermediate nodes acts as a selfish node and this selfishness is detected by Neighbor Monitoring Mechanism discussed in section 5.2.

5.1. Reducing over burden of sender

In SPIRITE, there is over burden on sender as discussed in section 3. To reduce the overburden of sender node, the intermediate nodes are assigned credits that follow a particular pattern wherein the node following the sender node is allotted a certain credit value α, and the subsequent nodes are given values a fixed amount β less than the previous ones. Thus the first node has value α, second node has α- β, third α-2β and so on. The n^{th} node will have the credit value α – (n-1) β as shown in figure 4.

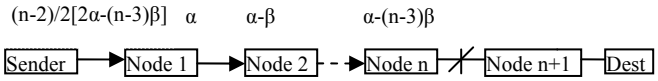

Figure 4: Illustration of payment scheme of MODSPIRITE

Here α is considered to be 1 and β is considered to be (α/2n). In the proposed scheme minimum credit to be assigned to forward data from source to destination is [(3/4n)(n-2)(n+1)]. The burden on sender becomes [3/4n((n-2)(n+1))] as node n+1 drops the packet.
On comparing both the SPIRIT and MODSPRITE, reduction of burden on sender is [(n^2 – 4.96n +6)/ (4n(n-1.99) * 100%. Figure 5 shows the reduced overhead on sender. If there are 10 nodes in a network then the burden on sender reduces by 17.6%, if there are 50 nodes in a network then the burden reduces by 23.5% and if there are 100 nodes a network then the burden reduces by 24.2%.

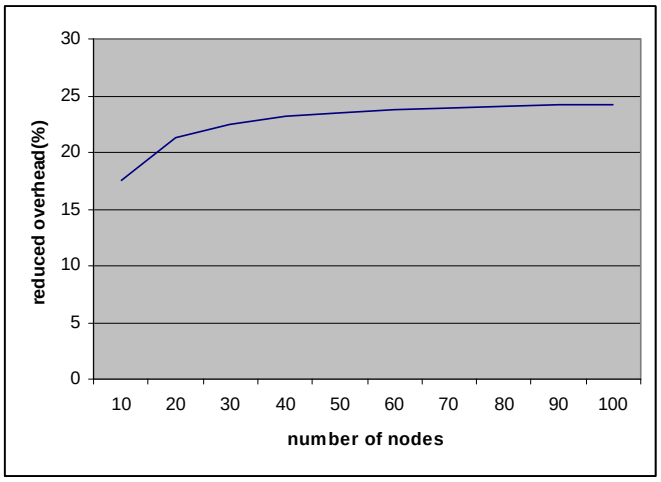

Figure 5: Reduced overhead (%) of sender

These indicates that the proposed scheme work better for larger network.

5.2. Stage of Ambiguity

Neighbor monitoring mechanism is applied to detect and to solve the stage of ambiguity as discussed in figure 2 in section 3. This mechanism is applied to few nodes only there is stage of ambiguity. For example in figure 2, mechanism is applied to node 3 and node 4 only. This will reduce overhead of calculating reputation in the entire network.

Neighbor Monitoring Mechanism

The neighbor monitoring mechanism is used to collect information about the packet-forwarding behavior of the neighbors. As the promiscuous mode assumes, a node is capable of overhearing the transmissions of its neighbors. With this capability, a mobile node n can maintain a neighbor node list (denoted by *NNL*) which contains its entire neighbor nodes information that node n learns about by overhearing. In addition, node n keeps track of two numbers for each of its neighbors (denoted by n+1), as described below:

• *R*(n+1): the total numbers of packets that node N has transmitted to n+1 for forwarding.

• *H*(n+1): the total number of packets that have been forwarded by n+1 and noticed by n.

The two numbers are updated according to the following methods. When node n sends a packet to node n+1 for forwarding, the counter *R*(n+1) is increased by one. Then n listens to the wireless channel and checks whether node n+1 forwards the packet as expected. If n detects that n+1 has forwarded the packet before a preset time-out expires, the counter *H*(n+1) is increased by one. If the value of both R(n+1) and H(n+1) are same then the node is said to be cooperative else it is selfish node and a punishment of γ is applied to the selfish node.

After detecting selfish node using neighbor monitoring mechanism, the payment scheme will be decided by the sender through cluster head. This will generate two cases.

Case I: Node n +1 is selfish and hence drops the message

Figure 6 shows the payment scheme when n+1th node is selfish. In this case sender gives credit of α to next node, α-β to the next node and so on. Thus the first node has value α, second node has α- β, third α-2β and so on. The n^{th} node will have the credit value α-(n-1)β. Thus sender has to pay a total of [n/2(2α-(n-1)β]. Here α is considered to be equal to 1, β is a very small value. γ credit is lost by node n+1 which drops packets and acts as a selfish node.

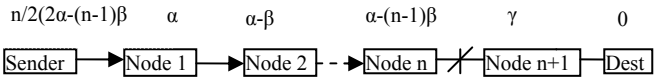

Figure 6: Illustration of payment scheme for Case I

Case II: Node n is selfish and hence drops the message

Figure 7 shows the payment scheme when nth node is selfish. In this case sender gives credit of α to next node, α-β to the next node and so on. Thus the first node has value α, second node has α- β, third α-2β and so on. The $n\text{-}1^{th}$ node will have the credit value α-(n-2)β. Thus sender has to pay a total of [n/2(2α-(n-2)β]. Here α is considered to be equal to 1, β is a very small value. γ credit is lost by node n which drops packets and acts as a selfish node. Punishment of γ on selfish node is given by sender and is indicated by the cluster head. This motivates other nodes to forward packet correctly.

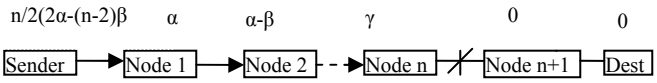

Figure 7: Illustration of payment scheme for Case II

6. Conclusions

Ad Hoc Networks have been an active area of research over the past few years. Such a network is highly dependent on the cooperation of all its nodes to effectively perform communication between nodes. This makes such a network highly vulnerable to selfish nodes.

This paper discusses on various reputation based and credit based mechanism to solve the problem of non cooperative nodes. Also, this paper proposes a credit based solution called MODSPIRITE to encourage cooperation among non cooperative nodes. This system is an improvement of

SPIRITE protocol. The burden on sender of losing credits is significantly reduced using MODSPRITE system. The MODSPRITE system reduces burden upto 24% for network that consists of 100 nodes. The neighbor monitoring mechanism is used to detect selfish nodes and applied to limited number of nodes; hence reducing the computing overhead. Punishment is given to the nodes that are unwilling to forward others' data. By penalizing such nodes make them motivated to forward others' node data.
There are certain privacy issues to be taken care of, prominent among them being preventing the security breach due to modification of credit value while transmission from the cluster head. The decision of which node should become the Cluster head is also a major issue. This issue must be taken care for future work.

References

[1] Elizabeth M. Royer, Chai-Keong Toh, "*A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks",* Proc. IEEE Personal Communication, March 1999, pp. 46-55.
[2] D. Johnson, D. Maltz, and J. Broch, "The dynamic source routing protocols for mobile ad hoc networks," *Internet Draft, IETF Mobile Ad- Hoc Network Working Group*, October 1999.
[3] Matthias Hollick, Jens Schmitt, Christian Seipl, and Ralf Steinmetz, "On the Effect of Node Misbehavior in Ad Hoc Networks" proc IEEE communication society,Vol.6 ,2004, pp 3759 - 3763
[4] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", In *Proceedings of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom2000)*, August 2000, pp 255–265.
[5] Pietro Michiardi and Refik Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks", Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia, 2002.
[6] S. Buchegger and J-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes, Fairness In Dynamic Ad-hoc Networks", Proc. of the IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), June 2002.
[7] Sonja Buchegger, Jean Yves Le Boundee, " Self – policing in Mobile Ad hoc Networks" In CRC Press, Chapter Handbook on Mobile Computing, December 2004.
[8] Tamer Refaei, Vivek Srivastava, LuizDaSilva, **"**A Reputation-based Mechanism for Isolating Selfish Nodes in Ad Hoc Networks", Proc. IEEE Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05), 2005.
[9] Fei Wang, Yijun Mo, Benxiong Huang,"COSR: Cooperative on Demand Secure Route Protocol in MANET", IEEE ISCIT, China, 2006.
[10] L. Buttyan and J. P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, MA, August 2000.
[11] S. Zhong, J. Chen, and Y. Yang, "Sprite: a simple, cheat-proof, credit based system for mobile ad-hoc networks," *IEEE INFOCOM*, San Francisco, CA, USA, April 2003.
[12] Upkar varshney, " Improving Wireless Health Monitoring Using incentive Based Router Cooperation" , In Proc. IEEE Computer Society, 2008, pp 56-62.
[13] Qi He, Dapeng Wu, Pradeep Khosla," SORI: A Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks", WCNC / IEEE Communications Society, 2004.
[14] Haiying Shen and Ze Li," ARM: An Account-based Hierarchical Reputation Management System for Wireless Ad Hoc Networks ,The 28th International Conference on Distributed Computing Systems Workshops, IEEE, 2008
[15] Hameed Janzadeh, Kaveh Fayazbakhsh, bahador bakshi," A secure credit-based cooperation stimulating mechanism for MANETs using hash chains", Future Generation Computer Systems -Elsevier 2009,pp 926-934

Rekha Kaushik holds a Master of Technology from Barkatullah University , Bhopal , M.P. India and pursuing Ph.d from Maulana Azad National Institute Of Technology(MANIT), Bhopal, India. She is a member of ISTE. Her general research interests include wireless communication especially Ad-hoc Networks and Network security.

Dr. Jyoti Singhai is Associate professor in Maulana Azad National Institute of Technology(MANIT), Bhopal, India. She holds Ph.D degree from MANIT, India. Her general research interests include wireless communication, image processing, and network security.