Volume 1 No. 2, June 2011
International Journal of Information and Communication Technology Research

http://www.esjournals.org

A Survey of Methods to mitigate Selfishness in Mobile Ad hoc Networks

[1]Sangheethaa Sukumaran, [2]Venkatesh. J, [3]Arunkorath
[1]Anna University Institute of Technology, Jothipuram Campus, Coimbatore
[2]Department of Management Studies, Anna university Institute of Technology,Coimbatore
[3]Department of CSE, Vedavyasa Institute of Technology, Karad PO, Malappuram Dt, Kerala
[1]sangheethaa@gmail.com, [2]venkijv@gmail.com , [3]arunkorath@vedavyasa.org

ABSTRACT

In recent years mobile ad-hoc networks have become very popular because of their widespread usage. Cooperation among the nodes in ad-hoc networks is an important issue for communication to be possible. But some nodes do not cooperate in communication and saves their energy. These nodes are called as selfish nodes. In the literature there are many methods which deal with the selfish behavior of the nodes. This paper compares different methods available for reducing the effect of selfish nodes in mobile ad hoc networks. This paper also presents some new approaches.

Keywords: *Ad hoc networks, selfish nodes, credit based, reputation*

1. INTRODUCTION

The emerging mobile ad hoc networking technology seeks to provide users "anytime" and "anywhere" services in a potentially large infrastructure less wireless network, based on the collaboration among individual network nodes. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. In such a dynamic environment routing the packets reliably to the destination becomes a critical issue.

All the nodes in an ad-hoc network acts as a router and cooperate among themselves for proper functioning of the network. It is assumed that all the nodes that participate in the network will do forwarding and routing in favor of other nodes. But this assumption does not work in all cases. Sometimes the nodes agree to forward, but fail to do because they want to save their battery power and CPU cycles. They just keep receiving the data destined to them, and drop the data of other nodes without forwarding or routing them, which reduces the throughput of the network. These nodes are called as Selfish nodes. They are classified under misbehaving nodes. There are another class of nodes which intentionally drop data, forward to different destination or misroute the data etc. They are called as malicious nodes. This paper deals with only selfish nodes.

Selfishness can be handled in two ways. One way is to punish the nodes for being selfish. Another way is to reward the nodes for not selfish. There are many approaches in the literature which follows either the first of the second method. This paper discusses the methods for mitigating the effect of selfishness and organized as follows. Next section gives overview of approaches for handling selfish nodes.

2. CREDIT BASED METHODS

Credit based methods are also called as incentive based methods. In these methods selfish nodes are not punished instead unselfish nodes are rewarded for helping other nodes. This stimulates the cooperation of nodes in the network. This section discusses some of the credit based systems in the literature.

2.1 Secure Incentive Protocol

Yanchao Zhang et al [1] inherited much of the features from [2] and [3]. This approach assumes that each mobile node (MN) has a tamper-proof security module such as SIM cards in GSM networks, which deals with security related functions and each intermediate node (IN) puts non-forged stamps on the forwarded packets as a proof of forwarding. Secure Incentive Protocol, (SIP) uses "credits" as the incentives to stimulate packet forwarding. For this purpose, each smartcard has a credit counter (CC) which is pre-charged with a certain amount of credits before shipped out. The charging and rewarding on a node is done by decreasing or increasing the CC in that node .And the CC will retain its value even when the MN is power off. When the MN is power-on again, it could still reuse the credits in the CC even in another SIP-enabled ad hoc network. To guarantee the security of SIP, each smartcard contains a private number and a public number (keys). The nodes have no knowledge about the keys stored in the smartcard and could not change CC in an unauthorized way either. SIP is session-based and mainly consists of three phases. During the first *Session initialization* phase, a session initiator (SI) negotiates session keys and other information with a session responder (SR) and INs between them. And each IN puts a non-forged stamp on each data packet forwarded and SI/SR collect those stamps for later rewarding use in the next *Data forwarding* phase. The final phase is R*ewarding* phase, in which each IN is awarded a certain number of credits based on the number of forwarded packets. Advantages of this method are 1. SIP is routing-independent in the sense that it could coexist with any on-demand unicast routing protocol such as DSR and AODV. 2. SIP is session based rather than packet based. 3. Security module is tamper proof and hence unauthorized access is not allowed. But the problem with this approach is, it needs every node to possess the hardware module and SIP is implemented in the hardware module. Hardware module will not be available in the already existing mobile nodes.

2.2 Stimulating Cooperation in Self Organizing Manets

L.Buttayan et al [3] focuses on packet forwarding and they address the problem of stimulating co-operation in self organizing Mobile Ad-hoc Networks for civilian applications. This approach uses a tamper resistant hardware module called "security module". This security module maintains a nuglet counter. When the node forwards a packet for the benefit of other nodes, the nuglet counter is increased by one, when it sends its own data the counter is decremented by one. Every node has to maintain a +ve counter value in order to send its own data. The nuglet counter is protected from illegitimate manipulations by the tamper resistance of the security module. This approach ensures that the misbehavior is not beneficial and hence it should occur rarely only. But the availability of hardware module is not guaranteed in general.

2.3 Sprite

Sprite, was proposed by Zhong et al. in [4]. In Sprite, nodes keep receipts of the received/forwarded messages. When they have a fast connection to a Credit Clearance Service (CCS), they report all these receipts. The CCS then decides the charge and credit for the reporting nodes. In the network architecture of Sprite, the CCS is assumed to be reachable through the use of Internet, limiting the utility of Sprite.

3. IDENTIFYING AND ISOLATING SELFISH NODES

This section explains methods that are used for punishing the selfish nodes. Selfish nodes are identified and isolated from the network. They are stopped from using the network services. Most of the approaches in the literature are following punishing system rather than rewarding system.

3.1 Watch Dog and Path Rater

S.Marti et al [5] addresses the problem of nodes agreeing to forward packets of other nodes but fail to forward. This describes two mechanisms to improve the throughput of the network. One mechanism is the watchdog, which identifies the misbehaving node by monitoring the nearby nodes whether they forward the packets of other nodes in the network. The other mechanism is the path rater that defines the best route by avoiding those misbehaving nodes. Since this approach tries to avoid the misbehaving nodes for routing, there's less chance of dropping packets, thus providing a better throughput even in the presence of high number of misbehaving nodes. But this approach does not isolate the misbehaving nodes; they still utilize the network services, i.e. the nodes are not punished for misbehaving.

3.2 CORE

Michiardi and Molva [6] proposed a Collaborative Reputation (CORE) mechanism that also has a watchdog component for monitoring. Here the reputation value is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request. In CORE the reputation value ranges from positive (+) through null (0) to negative (-). The advantage of this method is that having a positive to negative range allows good behavior to be rewarded and bad behavior to be punished. This method gives more importance to the past behavior and hence tolerable to sporadically bad behavior, e.g. battery failure. But the assumption that past behavior to be indicative of the future behavior may make the nodes to build up credit and then start behaving selfishly

.

3.3 CONFIDANT

CONFIDANT[7] collects evidence from direct experiences and recommendations. Trust relationships are established between nodes based on collected evidence trust decisions are made based on this relationships. There are four interdependent modules: (a) monitor, (b) reputation system, (c) path manager, and (d) trust manager. Monitor collects evidence by monitoring the transmission of a neighbor after forwarding a packet to the neighbor. It then reports to the reputation system only if the collected evidence represents a malicious behavior. Reputation system changes the rating for a node if the evidence collected for a node's malicious behavior exceeds the pre-defined threshold value. Then, path manager makes a decision to delete the malicious node from the path. Also path manager assists the node in making decision such as whether to forward a received packet by checking the upstream node's identity (previous-hop) in the blacklist. Trust manager is responsible for forwarding and receiving recommendations to and from trustworthy nodes. Here recommendations are known as ALARM messages and trustworthy nodes are referred as friends. The ALARM messages received from friends are evaluated for trustworthiness before being sent to the reputation system. Trust manager assists in making trust decisions for the following, whether to: (a) provide and accept routing information, (b) accept a node as a part of route, and (c) take part in a route originated by some other node.

CONFIDANT proves to show better network performance in presence of malicious nodes compared to DSR protocol.

3.4 Cache scheme to detect Selfish nodes

Hongxun Liu et al [8] proposed a hardware assisted detection scheme is proposed and evaluated. In this scheme, the hardware can detect the misbehavior conducted by the selfish nodes. Selfish node either drops all the packets not related to it or drops the data packets only. Upon detecting the misbehavior, the hardware will report the misbehaving node (itself) to other nodes. The other nodes will use the information received to protect the network. In this scheme, there is a separation between software and hardware inside a single mobile node. The software could be misbehaving, but the hardware is tamper resistant and is the cornerstone of building trust relationship among mobile nodes. Here the focus is on the detection of misbehaving node dropping packet. There are two kinds of packet dropping conducted by the misbehaving nodes, simple dropping and selective dropping. In simple dropping, the misbehaving nodes will drop all the packets not to or from them; while in selective dropping, the misbehaving nodes will only drop data packets not to or from them while forwarding the control packets, such as route request, route reply, etc. There are four counters used in the cache based detection scheme: TC (Total Counter), DC (Drop Counter), TDC (Total Data Counter) and DDC (Data Drop Counter). The first two counters are used to detect simple dropping while TDC and DDC are used to detect selective dropping. TC is used to record the total number of unique packets received, while DC is used to record how many unique packets are dropped by this node. TDC is used to record how many data packets are received by the node while DDC records the number of data packets dropped. Another timer is added to improve the detection performance. The timer is used to give additional penalty if a node doesn't forward a route request. The penalty timer (PeT) is started when an original route request is received. If the node doesn't forward the route request during the period of PeT, an extra penalty is added to DC. PeT is only started when an original route request is received. A duplicate route request will not initiate PeT. The cache unit inside the detection hardware can tell if a received route request is original or duplicate. The authors also argue that the cache scheme can detect the misbehaving nodes accurately in terms of detection effectiveness and false positive in both the simple dropping and the selective dropping scenarios. Also only minor changes are needed in the software layer. But still it is a hardware based approach. Thus has its own practical difficulty in implementation.

3.5 Acknowledgement Based Schemes

3.5.1 2ACK Scheme over DSR

Kejun Liu et al [9] considers only packet forwarding misbehavior. When a node forwards data packet success fully over the next hop, the destination node of next hop will send back a special two hop acknowledgement called 2ACK. This method works along with DSR[10] protocol. There are many disadvantages in this approach. This paper does not address what happens when a 2ACK got lost or dropped by a malicious node, or what happens if a malicious node sends the 2ACK packet without forwarding the data packets. i.e the node does not forward t he data packet, but it simply sends 2ACK which act as an acknowledgement for the 2 hop neighbor.

3.5.2 ACK Scheme over AODV Protocol

T.V Sundararajan et all [11] proposes a method which also follows 2ACK scheme but works on AODV [10] protocol. It follows the same concepts of 2ACK scheme. But the acknowledgements will anyway increase the overhead in the network. Also there are chances for false positives. i.e a well behaving node may be considered as misbehaving. This paper does not deal with loss of acknowledgements.(2ACKs).

3.5.3 Improved Acknowledgement Based Scheme to detect packet dropping attack

Aishwarya Sagar et al [12] classify selfish nodes into 3 types as in [13]. (i.e. SN1, SN2 and SN3). SN1 nodes take participation in the route discovery and route maintenance phases but refuses to forward data packets to save its resources. SN2 nodes neither participate in the route discovery phase nor in data-forwarding phase. Instead they use their resource only for transmissions of their own packets. SN3 nodes behave properly if its energy level lies between full energy-level E and certain threshold T1. They behave like node of type SN2 if energy level lies between threshold T1 and another threshold T2 and if energy level falls below T2, they behave like node of type SN1. Here each node maintains a LIST which consists of ID of every data packets sent or forwarded. After forwarding data packet to the next hop along the active route, LNode of every group will make an entry of forwarded data packet in the LIST and wait for ACK-1 and ACK-2 packet which are sent from RNode of first set and RNode of second set respectively. Also ACK-1 and ACK-2 packet must be received within time T1 and T2 respectively. There are 3 steps. 1. Detection of malicious group - Before identifying malicious or misbehaving node, network should be aware that some malicious activity is present or not2. Identification of particular misbehaving node- Based on whether the acknowledgement is received within the time limit or not. 3. Isolation and mitigation of misbehaving node –by avoiding the detected misbehaving node and updating LIST of misbehaving nodes. A comparison with other acknowledgement based scheme is available in [12].

3.5.4 A Robust Approach to Detect and Prevent Network Layer Attacks

Volume 1 No. 2, June 2011
International Journal of Information and Communication Technology Research

http://www.esjournals.org

The algorithm designed in [14] mainly identifies four attacks parallelly namely, packet eavesdropping, message tampering, black hole attack and gray hole attack. These attacks are identified by setting different threshold values to the ratio Cmiss/Cpkt where Cmiss represents number of packets lost and Cpkt is the number of packets sent. If the ratio calculated exceeds the limit of tolerance threshold value 20%, then the link is said to be misbehaving otherwise properly behaving. Parallelly using the ratio value, the corresponding attacks will be identified. The algorithm looks simple but setting up the threshold value to 20% or any other percentage needs further clarifications.

3.6 Methods based on routing protocols

There are some approaches which identify and isolate misbehaving nodes by modifying the existing routing protocols for ad hoc networks. This section discusses some of those approaches.

3.6.1 Extended DSR

V. Narasimha Raghavan et al [15] modified the existing Dynamic Source Routing protocol based on the extent of friendship between the nodes to make the nodes to co-operate in an ad hoc environment. Here a node classifies its neighbors as a stranger – if there was no communication between them, acquaintance – communicated for some time or a friend- if communicated several times. Based on this classification trust level is established as" no trust, low trust or high trust" Each node maintains a friendship table showing the relationship of one node with its neighbors. When a node wants to communicate with other node, route request is sent as a broadcast to all its neighbors. Route reply obtained from its neighbor is sorted by trust ratings. The source selects the most trusted path. If it's one hop neighbor node is a friend, then that path is chosen for message transfer. If a node is found to be selfish its packets are not forwarded thus isolating the selfish nodes from the network.

3.6.2 Trust based Secure Routing Protocol

Houssein Hallani and Seyed A. Shahrestani [16] proposed a fuzzy based trust model for nodes. This approach works on AODV routing protocol. Fuzzy logic helps to quantify trust between nodes in ad hoc networks. This paper addresses the following problems. Packets dropped, wrong forwarding, fabrication and replay attacks. This evaluation model is a Mamdani type with four input and one output variables. The elements of a fuzzy set are mapped by membership functions to a value, which defines the degree to which a fuzzy variable is a member of a set. The membership functions $\mu(P)$, $\mu(WF)$, $\mu(F)$, $\mu(RA)$, $\mu(T)$, map the input variables, packet_dropped, wrong_forwarding, fabrication and replay_ attack, and the output variable, trust_level, into the interval (0,1) respectively. After applying the fuzzy trust evaluation model each node will have a trust level. Each node is assumed to be able to evaluate the trust level of each of its neighboring nodes based on the information regarding the behavior history of these nodes. These trust levels are then used to determine the most appropriate route between S and D. But this approach is specific for AODV [17]. Also, mapping the trust level using fuzzy trust evaluation model itself is energy consuming.

3.6.3 Local Detection of Selfish nodes

Bo Wang et al [18] in their paper used a finite state machine model of locally observed AODV actions to build up a statistical description of the behavior of each neighbor. They applied a series of well known statistical tests to features derived from this description to partition the set neighboring nodes into a cooperative and selfish class. This approach detects both route request drops and route reply drops by the selfish nodes. Selfish behavior is distinguished from cooperative behavior by comparing the statistical behavior of neighbors across multiple local routing instances.

3.7 Reputation Based Access Control Mechanisms

Some approaches in the literature follow reputation based mechanism which can be either combined with any of the routing protocol or used as a separate framework.

3.7.1 Reputation Based Intrusion Detection System

Animesh KR Trivedi et al [19] proposed a reputation based intrusion detection system for Mobile Ad-hoc Networks (RISM). RISM system runs on every node in network and consists in core of the following modules: The Monitor, holds the responsibility of monitoring activities in the Neighborhood using PACKs (Passive ACKnowledgements) Every node registers all the data packets sent by it to next node and when it receives packets in promiscuous mode, it matches those to the queue of registered packets present in its buffer. After a fixed time interval -termed as the Timing Window, nodes make a log of number of packets for which they haven't received acknowledgment in the form of PACK and communicate it to the reputation manager. Monitor maintains a log of activity of next neighbor for each Window and sends it to Reputation manager. Reputation system receives activity log of next hop neighbor from monitor with number of packets for which it does not receive PACK, called as Missing or Dropped Packets. The number of missing packets is then compared with the Malicious Drop Threshold and if it is comparatively lesser, then the reputation manager gives positive performance appraisal else negative. The path manager performs trivial path management functions in collaboration with DSR core.

Redemption and Fading are included in design of RISM to allow nodes previously considered malicious to become part of network again as ad-hoc networks run on cooperation and collaboration of peer nodes and no one gets benefited without cooperating with each other. Congestion parameter, Knock test and Timing window are some new concepts that are introduced in this paper.

3.7.2 Reputation Based mechanism to isolate selfish nodes

M. Tamer Refaei et al [20] proposed reputation-based mechanism as a means of building trust among nodes. The mechanism relies on the principle that a node autonomously (i.e., without communicating with other neighboring nodes) evaluates its neighbors based on the completion of the requested service(s). This mechanism based on trust management schemes does not rely on the monitoring of neighbors' transmissions and the exchange of reputation information among nodes thus involves less overhead, and this approach does not rely on any routing protocol. This approach provides a distributed reputation evaluation scheme implemented autonomously at every node in an ad hoc network with the objective of identifying and isolating selfish neighbors. Each node maintains a reputation table, where a reputation index is stored for each of the node's immediate neighbors. A node ascribes a reputation index to each of its neighbors based on successful delivery of packets forwarded through that neighbor. For each successfully delivered packet, each node along the route increases the reputation index of its next-hop neighbor that forwarded the packet. Conversely, packet delivery failures result in a penalty applied to such neighbors by decreasing their reputation index. In other words, when a node transmits a packet to one of its neighbors, it holds the neighbor responsible for the correct delivery of the packet to the final destination. The indication of a success or failure is obtained from feedback received from the destination (e.g., using TCP acknowledgements). The function used to compute the reputation index is a design decision that is influenced by factors including node behavior, node location, as well as others.

To prevent selfish behavior and to provide motivation for nodes to build up their reputation, each node determines whether to forward or drop a packet based on the reputation of the packet's previous hop. Once a node's reputation, as perceived by its neighbors, falls below a pre-determined threshold all packets forwarded through or originating at that node are discarded by those neighbors and the node is isolated. Advantages of this approach are, 1. Routing Protocol independence, 2. No need for monitoring the neighboring nodes in a promiscuous mode. 3. Less overhead since nodes does not pass reputation information. 4, Reputation value is calculated without the help of neighbors, so collaborative misbehavior can be mitigated. But the problem with this approach is that, it uses feedback mechanisms like TCP acknowledgements in connection oriented applications for identifying whether a packet has reached the destination or not. So this method is not suitable for connectionless applications. The reputation mechanism independently calculates the behavior of the nodes, so there are more chances for false positives.

3.8 Ticket Based Approaches for Access Control

There are many approaches in the literature, which deals with selfish nodes in ad-hoc networks using tickets.

3.8.1 Centralized and Distributed server

L.Zhou et al [4] and G.Appenzeller et al [21] proposed ticket based approaches. Tickets are provided for the nodes, which are well behaving, and network access is provided only to the nodes with a valid ticket. The ticket is obtained from a centralized authority [22] or from distributed servers [23]. The central server approach has several advantages and disadvantages. The central server approach can work well for a simple, less dynamic network. But for a dynamic network the delay will be more. The distributed approach has no much difference with central authority system except that here there are three or more central servers in the network. In both the approaches when the central server fails, the network functioning becomes vulnerable to attacks.

3.8.2 Localized Approach for Access Control

The localized approach for access control is proposed by Haiyoun Luo et al [24]. This is a ticket-based approach. The localized approach proposes a fully localized design paradigm to provide ubiquitous and robust access control for mobile ad hoc networks. Each well behaving node uses a certified ticket to participate in routing and packet forwarding. Nodes without valid tickets are classified as misbehaving. They will be denied from any network access, even though they move to other locations. Thus, misbehaving nodes are "isolated" and their damage to the mobile ad hoc network is confined to their locality. The access control operation emphasizes multiple node consensus and fully localized instantiation. Since any individual node is subject to misbehaviors, this approach does not rely on any single node. Instead, the nature of cooperative computing in an ad hoc network is leveraged and the approach depends on the collective behaviors of multiple local nodes. Here multiple nodes in a local network neighborhood, typically one or two hops away, collaborate to monitor a node's behavior and determine whether it is well-behaving or misbehaving using certain detection mechanism of their choice. These local monitoring neighbors will renew the expiring ticket of a well behaving node collectively, while a misbehaving node will be denied from ticket renewal or be revoked of its ticket. In this way, the functionality of a conventional access control authority, which is typically centralized, is fully distributed into each node's locality. Every node contributes to the access control system through its local efforts and all nodes collectively secure the network.

The localized approach does not need any hardware module for security. It does not assume anything about the packet size or type of traffic or the type of data. It not only detects the misbehaving nodes but also isolates them from the network. Average delay for ticket renewal is tolerable, because the node gets its ticket from its locality rather than going to a central server. There's no necessity for the node to rely upon a single node for getting a ticket or for renewal. So this approach is highly robust and scalable.

The localized approach requires that each node should get k tickets from its local neighborhood. It is possible to get k number of tickets in a highly populated network. But it is not possible when the number of nodes in a network is less. Thus the localized approach cannot be used in a sparse network. Moreover the protocol used in localized approach broadcasts the ticket request to all its neighbors, which increases the communication overhead. The efficiency of the localized approach depends upon the coalition size k. i.e. the number of partial tickets that the node should get to access the network. The parameters viz. average delay, overhead and success ratio, which are used for simulation in [24], vary depending upon the k value. The k value is fixed as 5 in [24] based on the network size. This value does not change when number of nodes in the network increases or decreases. But this value will not work for all the networks. It is applicable only to a large network. For a sparse network, collecting 5 tickets from the neighborhood will cause more delay, because the nodes may not have sufficient number of neighbors in their locality. So in order to reduce the number of tickets a node should receive before successful access of the network, reputation mechanism can be used.

3.8.3 Reputation Based Localized Access Control

This paper [25] proposes a ticket-based approach, which uses reputation mechanism for evaluating the tickets. The nodes can access the network if they have a valid ticket. The tickets are obtained from the neighboring nodes, which have high reputation value. Initially the tickets are issued by a dealer. The tickets have expiration time. Once the expiration time reaches, the nodes have to renew their tickets. For renewing, the nodes will send the broadcast request to all its one-hop neighbors. On receiving the ticket renewal request, the neighbors have to decide whether to send a ticket or not by checking the reputation value of that node. Each node maintains the reputation value, by monitoring their behavior using any monitoring mechanism. When the requesting node receives a reply ticket, it checks the reputation value of the node, which has sent the reply. If the reputation value of the node is greater than a threshold value (this value is chosen based on the network behavior) then the requesting node accepts the ticket, otherwise it rejects the ticket from that node and looks for other replies. Once it receives a ticket from higher reputation node, the node uses that ticket to prove its behavior and access the network. This makes the ticket obtaining process simpler. Whenever a node issues a network access request, its ticket and the reputation value of the node, which gave the ticket, is verified. This ensures that two nodes cannot collaborate with each other and generate false tickets. Moreover other nodes will also monitor the behavior of these nodes. Nodes may try to generate their own tickets for communication. But this will be identified because the tickets are signed and verified using RSA algorithm. So this method is false proof and secure.

3.9 A Framework for Detection of Selfishness

This Paper [26] describes a new framework based on Dempster-Shafer theory-based selfishness detection framework (DST-SDF) with some mathematical background and simulation analysis. The DST-SDF is dedicated for MANETs based on standard routing like dynamic source routing (DSR). The main concept relies on end-to-end packet acknowledgments in the following way: every time a source node sends a packet to a destination node, it waits for a certain predefined time for an acknowledgement of the packet. If one arrives within the predefined time, the source node has reason to claim that all nodes on the path are cooperative (none is selfish). Otherwise if there are no other indications of faultiness on the path (e.g., RERR messages), the source node knows that there are selfish nodes on the path. Whenever an acknowledgment does or does not arrive in time, a special *recommendation message* is sent out to inform the other nodes about the detected situation (selfish or cooperative behavior on the path, respectively). Every node in the network is equipped with a dedicated component executing a DST-based algorithm that uses received recommendation messages to evaluate the selfishness of each node. The resulting values can be used as routing metrics while selecting packets' routes in the near future.

4. CONCLUSION

This paper discussed several approaches for dealing with selfish nodes. Selfish nodes are a real problem for ad hoc networks since they affect the network throughput. Many approaches are available in the literature. But no approach provides a solid solution to the selfish nodes problem. The Credit based approach provides incentives to the well behaving nodes and just by passes the selfish nodes in selecting a route to the destination. But selfish node still enjoys services without cooperating with others. The detection and isolation mechanism isolates the selfish nodes so that they don't receive any services from the network. Thus penalizing the selfish nodes. But what happens if many nodes become selfish? Network communication itself will become impossible. Thus we cannot eliminate all the selfish nodes from the network. A new method to reduce the effect of selfishness and stimulating the nodes to cooperate in the network services should be developed. But the overhead in achieving this should also be less. Because we should remember that after all we are dealing with battery operated devices.

REFERENCES

[1] Yanchao Zhang , Wenjing Lou , Wei Liu, Yuguang Fang, " A secure incentive protocol for mobile ad hoc networks" in Journal of Wireless Networks , Volume 13 Issue 5, pp. 663-678 , October 2007

[2] L. Buttyan and J.P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs", in Proc. of IEEE/ACM MobiHoc, Boston, Aug. 2000.

[3] L. Buttyan and J.P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," ACM Journal for Mobile Networks (MONET), Vol. 8, No. 5, Oct. 2003.

[4] S.Zhong, J.Chen, and Y.R.Yang, " Sprite: A Simple, Cheat Proof, Credit based System for Mobile Ad Hoc Networks", in Proceedings of INFOCOM, Apr. 2003.

[5] S.Marti, T.Giuli, K.Lai and M.Baker, "Mitigating Routing Misbehavior in Mobile Ad-Hoc Networks," in Proc. ACM MOBICOM, pp. 255-265,2000.

[6] 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.

[7] Buchegger, Sonja ; Le Boudec, Jean-Yves, " Performance Analysis of CONFIDANT Protocol: Cooperation of Nodes - Fairness in Dynamic Ad-Hoc Networks," in Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC). IEEE, June 2002.

[8] Hongxun Liu, José G. Delgado-Frias, and Sirisha Medidi, "Using a cache scheme to detect selfish nodes in mobile adhoc networks " in proceedings of IEEE international Conference on Networks, pp- 7 – 12, Nov. 2007

[9] Kejun Liu, Jing Deng, Pramod K. Varshney, and Kashyap Balakrishnan "An Acknowledgment-based Approach for the Detection of Routing Misbehavior in MANETs", in IEEE Transactions on Mobile Computing , Volume 6 Issue 5, May 2007

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, vol. 353, pp. 153–181, 1996.

[11] T.V.P. Sundararajan, Dr.A.Shanmugam, "Performance analysis of selfish node aware routing protocol for Mobile Ad-Hoc Networks" in ICGST ICNIR Journal, volume 9, Issue 1, July 2009.

[12] Aishwarya Sagar Anand Ukey1, Meenu Chawla "Detection of Packet Dropping Attack Using Improved Acknowledgement Based Scheme in MANET" in IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 1, July 2010.

[13] Abdelaziz Babakhouya, Yacine Challal, and Abdelmadjid Bouabdallah, "A Simulation Analysis of Routing Misbehaviour in Mobile Ad Hoc Networks," in Proc. of the Second International Conference on Next Generation Mobile Applications, Services, and Technologies, September 2008, pp. 592-597.

[14] G. S. Mamatha ,S. C. Sharma A Robust Approach to Detect and Prevent Network Layer Attacks in MANETS " in International Journal of Computer Science and Security Volume: 4 Issue: 3 Pages: 265-372 Publication Date: July 2010 ISSN (Online): 1985-1553.

[15] V. Narasimha Raghavan, T. Peer Meera Labbai, N. Bhalaji, and Suvitha Kesavan "Extended Dynamic Source Routing Protocol for the Non Co-Operating Nodes in Mobile Adhoc Networks" in International Journal of Applied Mathematics and Computer Sciences 3;1 © www.waset.org Winter 2007.

[16] Houssein Hallani and Seyed A. Shahrestani," Mitigation of the Effects of Selfish and malicious Nodes in Ad-hoc Networks" in WSEAS TRANSACTIONS on COMPUTERS Issue 2, Volume 8, PP.No206- 221, ISSN: 1109-2750,February 2009

[17] C.E. Perkins and E.M.Royer, " Ad hoc On-Demand Distance Vector Routing", in Proceedings of the 2nd IEEE workshop on Mobile Computing Systems and Applications., New Oreleans, LA, Feb.1999.

[18] Bo Wang Sohraab Soltani Jonathan K. Shapiro, Pang-Ning Tan " Local Detection of Selfish

Routing Behavior in Ad Hoc Networks" in proceedings of 8th International Symposium on Parallel Architectures,Algorithms and Networks, Dec. 2005

[19] Animesh Kr Trivedi1, Rishi Kapoor1, Rajan Arora1, Sudip Sanyal1 and Sugata Sanyal , " RISM - Reputation Based Intrusion Detection System for Mobile Adhoc Networks" Available from lin profile.iiita.ac.in/aktrivedi_b03/rism.pdf

[20] M. Tamer Refaei, Vivek Srivastava, Luiz DaSilva, Mohamed Eltoweissy, " A Reputation-based Mechanism for Isolating Selfish Nodes in Ad Hoc Networks",in Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05) , 2005

[21] S.Zhong, J.Chen, and Y.R.Yang, " Sprite: A Simple, Cheat Proof,Credit based System for Mobile Ad Hoc Networks", in Proceedings of INFOCOM, Apr. 2003.

[22] L.Zhou, E.B.Schnder and R.Can Renese "COCA: a Secure Distributed on Line Certificate Authority," ACM Tran, Computer Sys, vol. 2, no. 4,pp. 329-368, Nov. 2002.

[23] G.Appenzeller, M.Roussopoulous, and M.Baker, "User-friendly access control for public network ports," in Proc. IEEE INFOCOM, pp. 699-707, 1999.

[24] Haiyoun Luo, P.Zerfos, Songwu Lu, L.Zhang "URSA-Ubiquitous and Robust Access Control for Mobile Ad hoc Networks," IEEE/ACM Transactions on Networking, vol. 12, no. 6, pg. 1049-1063, Dec. 2004.

[25] Sangheethaa Sukumaran, Elijah Blessing, "Reputation Based Localised access control for mobile ad-hoc networks", in Lecture notes in Computer Science, Volume 4104/2006, ISSN 0302-9743.

[26] Jerzy Konorski and Rafał Orlikowski "A Framework for Detection of Selfishness in Multihop Mobile Ad Hoc Networks " in Jounral of telecommunications and information technology, pp 34- 40, 2009.