CHAPTER II

2. LITERATURE SURVEY

There are many approaches in the literature which deal with misbehaving nodes in MANETs. This chapter explains some of them in detail. For study purpose, the methods in the literature are divided into several classifications. They are; a) Credit Based Approaches or Incentive Based Approaches, b) Ticket Based Approaches, c) Secure Routing Protocols, d) Reputation based Approaches e) Acknowledgement Based Approaches and f) Mobile agent based approaches. In the next few sub sections each approach is explained in brief.

2.1 Credit Based Approaches or Incentive Based Approaches

This section explains credit based or incentive based approaches.

2.1.1 Auction based Ad-hoc On-demand Distance Vector Protocol.

Cenker Demir and Cristina Comaniciu (2007) have proposed an algorithm which is based on credit system. In order to deal with selfishness, digital economy is created in the network. In this virtual economy, a source node has to pay some amount of a digital currency to intermediate nodes to have its packet forwarded, whereas the intermediate nodes bid and declare the amount of currency that they would request from the source if they forward the packet. From these bids, source node chooses the route with the lowest bid. The source node sends the payment with every packet. Payment is set in a way that every node gets a payment that is greater than the amount that it bid.
When nodes bid, they consider their energy level and the amount of currency that they have. Their bid increases when their energy level goes down, and decreases when their currency level goes down. Consequently, an auction based routing mechanism will help to deal with selfish nodes in the network, and also help to increase fairness in distributing the energy consumption in the network.

2.1.2 Stimulating Cooperation in Self Organizing MANETs

L. Buttayan and J.P. Hubaux (2003) focused 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 misbehaviour is not beneficial and hence it should occur rarely only. But the availability of hardware module is not guaranteed in general.

2.1.3 Sprite

Sprite was proposed by Zhong et al (2003). Sprite is a simple, cheat-proof, credit based system for stimulating cooperation among selfish nodes in mobile ad-hoc networks. This system provides incentive for mobile nodes to cooperate and report actions honestly. 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. When a node sends its own messages, the node (or the destination, see later) will lose credit (or virtual money) to the network because other nodes incur a cost to forward the messages. On the other hand, when a node forwards others’ messages, it should gain credit and therefore be able to send its messages later. In the network architecture of Sprite, the CCS is assumed to be reachable through the use of Internet, limiting the utility of Sprite.

2.2 TICKET BASED APPROACHES

This Section explains some of the Ticket based approaches for dealing with misbehaving nodes.

2.2.1. Centralized and Distributed server

Zhou. L et al., (2002) and Appenzeller. G et al., (1999) 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 or from distributed servers. 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.
2.2.2. Localized Approach for Access Control

The localized approach for access control is proposed by Haiyoun Luo et al., (2004). 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 misbehaviour, 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 behaviours of multiple local nodes. Here multiple nodes in a local network neighbourhood, typically one or two hops away, collaborate to monitor a node’s behaviour and determine whether it is well-behaving or misbehaving using certain detection mechanism of their choice. These local monitoring neighbours 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 is 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 neighbourhood. 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 neighbours, 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 are used for simulation. They vary depending upon the ‘k’ value. The ‘k’ value is fixed as 5 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 neighbourhood will cause more delay, because the nodes may not have sufficient number of neighbours in their locality. The number of tickets that a node should receive before accessing the network can be reduced using reputation mechanism.

2.2.3. Reputation Based Localized Access Control

Sangheetha Sukumaran and Elijah Blessing (2006) proposed 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 neighbouring 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 neighbours. On receiving the ticket renewal request, the neighbours 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 behaviour 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 behaviour) 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 behaviour 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 behaviour 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.

2.3 SECURE ROUTING PROTOCOLS

This section explains some of the routing protocols used in the MANETs to provide security from misbehaving nodes.

2.3.1 Authenticated Routing for Ad-hoc Networks

Authenticated Routing for Ad-hoc Networks (ARAN) was proposed by Sanzgiri. K et al., (2005). It is an on-demand secure routing
protocol which relies on digital certificates. By using digital certificates, it provides authentication, message integrity and non repudiation. Thus it provides end to end guarantee during message delivery between source and destination. It is a secure routing protocol which provides solutions for routing attacks. This protocol considers only malicious nodes and not selfish nodes. Moreover, the computational cost involved is high compared to other related approaches. It also needs a secure certificate issue server.

2.3.2 Reputed ARAN

Abdalla Mahmoud (2005) proposed a modified ARAN by adding reputation mechanism to the existing ARAN. Reputed-ARAN is capable of handling both selfish and malicious nodes’ attacks. Each node keeps only the reputation values of all direct nodes it dealt with. These reputation values are based on the node’s first hand experience with other nodes. This protocol can also be grouped under reputation mechanisms.

Here, the source node sends Route Request as a broadcast. The destination replies to every route replies. Source selects a best path based on reputation and uses it for transmitting data. Destination gives acknowledgement for every data packet received successfully and the source updates the reputation table by giving a positive recommendation (+1) to the first hop of the reverse path. All the intermediate nodes in the route give a recommendation of (+1) to their respective next hop in the route and update their local reputation tables. If there is a selfish node in the route, the data packet does not reach its destination. As a result, the source does not receive any DACK for the data packet in appropriate time. So, the source gives a recommendation of (-2) to the first hop on the route. The intermediate nodes also give a recommendation (-2) to their next hop in the route up to the node that dropped the packet. As a consequence, all the nodes between the selfish
node and the sender, including the selfish node, get a recommendation of (-2). The idea of giving (-2) to selfish nodes per each data packet dropping is due to the fact that negative behaviour should be given greater weight than positive behaviour. In addition, this way prevents a selfish node from dropping alternate packets in order to keep its reputation constant. This makes it more difficult for a selfish node to build up a good reputation to attack for a sustained period of time. Moreover, the selfish node will be isolated if its reputation reached a threshold of (-40). This scheme has an obvious advantage of isolating selfish nodes in the network. But use of digital certificates as in ARAN causes more communication overhead. Here an acknowledgement packet called as DACK has to be generated for each data packet that it successfully receives. This also causes extra overhead in the network. Again the same problems faced in ARAN are applicable here.

2.3.3 Ariadne

Yih-chun Hu et al., (2005) in his paper proposes Ariadne a secure on-demand routing protocol. Ariadne prevents attackers or compromised nodes from tampering with uncompromised routes consisting of uncompromised nodes, and also prevents many types of Denial-of-Service attacks.

This paper classifies attacks on an ad-hoc network routing protocols into one of two categories: routing disruption attacks and resource consumption attacks. In a routing disruption attack, the attacker attempts to cause legitimate data packets to be routed in dysfunctional ways. In a resource consumption attack, the attacker injects packets into the network in an attempt to consume valuable network resources such as bandwidth, or to consume node resources such as memory (storage) or computation power.
In addition, Ariadne is efficient, using only highly efficient symmetric cryptographic primitives and it is based on TESLA concept. TESLA is an efficient authentication scheme that requires loose time synchronization. Firstly it verifies route authenticity and secondly it checks that no node is missing on RREQ message. It is vulnerable to an attacker that happens to be along the discovered route. Ariadne needs the security association between the initiator to every node including intermediate node and the source node. Ariadne prevents attackers with uncompromised routes and also prevents many types of Denial-of-Service attacks. This protocol mainly deals with attacks from the application layer perspective.

2.3.4 MARS: Misbehaviour Detection in Ad-hoc Networks

To detect misbehaviour on data and mitigate adverse effects, Li Zhao (2007) proposed Multipath Routing Single path transmission (MARS) scheme. The MARS combines multipath routing, single path data transmission, and end-to-end feedback mechanism together to provide more comprehensive protection against misbehaviour from individual or cooperating misbehaving nodes. It is assumed that the required paths, which may not be free of misbehaving nodes, between end nodes have been discovered by routing protocols. The source selects two node-disjoint paths: one is used for data transmission; the other is for transmission information exchange. The destination detects misbehaviour and notifies source through the feedback mechanism. The adverse effects are mitigated without restrictive assumptions on the network nodes’ trust, without the use of intrusion detection schemes, and at the expense of moderate overhead only.

This paper deals with misbehaving of nodes on data like dropping the data, modifying the data, colluded dropping etc. The MARS scheme tackles misbehaviour through the use of two new types of control
packets, termed INF and NTF. An INF packet, used to detect misbehaviour, is sent from the source to the destination at the start of data transmission. A NTF packet, used to mitigate the adverse effects, is sent from the destination to the source when suspected misbehaviour along data transmission path is detected. The authors claim that this approach could effectively detect individual and colluding misbehaviour. But use of multipath routing is not effective in case of sybil attack in which a single node presents multiple identities to other nodes.

2.3.5 Reactive identification of misbehaving nodes based on random audits

This approach proposed by Kozma, W. and Lazos, L (2008), identifies only the misbehaving nodes which selectively drop packets. This method does not depend on energy intensive overhearing techniques or communication intensive acknowledgement services. It checks behaviour of the node, only when the network throughput reduces below a threshold level hence called as reactive. Random audit is initiated when the destination notifies the source about the poor path performance thus significantly reducing the associate resource overhead. Source node audits whether intermediate node has forwarded the packet P_i randomly using bloom filter (details of bloom filter not discussed here). If nodes are selfish, dropping all packets, this approach will not work. Also the assumption that there are at least two independent paths to any destination i.e., the network is two-connected may not work in all cases.

2.3.6. Watch Dog and Path Rater

S. Marti et al., (2000) 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 is 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.

2.3.7. Trust based Secure Routing Protocol

Houssein Hallani and Seyed A. Shahrestani (2009) 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 neighbouring nodes based on the information regarding the behaviour 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. Also, mapping the trust level using fuzzy trust evaluation model itself is energy consuming.
2.4. REPUTATION BASED SYSTEMS

This section deals with some of the reputation based approaches for dealing with misbehaving nodes.

2.4.1 Reputation Based Intrusion Detection System

Animesh KR Trivedi et al., (2007) proposed a reputation based intrusion detection system for Mobile Ad-hoc Networks (RISM). RISM system runs on every node in network and consists of the following modules in core: The Monitor holds the responsibility of monitoring activities in the neighbourhood 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 neighbour for each window and sends it to Reputation manager. Reputation system receives activity log of next hop neighbour 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 a positive performance appraisal else negative. The path manager performs trivial path management functions in collaboration with DSR core. Redemption and Fading are included in the 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. Even though this method is classified under reputation based approach, correctness of the system mainly depends on the acknowledgements.

2.4.2 Reputation Based mechanism to isolate selfish nodes

M. Tamer Refaei et al. (2005) 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 neighbouring nodes) evaluates its neighbours based on the completion of the requested service(s). This mechanism based on trust management schemes does not rely on the monitoring of neighbours' 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 neighbours. Each node maintains a reputation table, where a reputation index is stored for each of the node’s immediate neighbours. A node describes a reputation index to each of its neighbours based on successful delivery of packets forwarded through that neighbour. For each successfully delivered packet, each node along the route increases the reputation index of its next-hop neighbour that forwarded the packet. Conversely, packet delivery failures result in a penalty applied to such neighbours by decreasing their reputation index. In other words, when a node transmits a packet to one of its neighbours, it holds the neighbour 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 behaviour, node location, as well as others.

To prevent selfish behaviour 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 neighbours, falls below a pre-determined threshold all packets forwarded through or originating at that node are discarded by those neighbours and the node is isolated.

Advantages of this approach are,
1. Routing Protocol independence,
2. No need for monitoring the neighbouring nodes in a promiscuous mode,
3. Less overhead since nodes does not pass reputation information, and
4. Reputation value is calculated without the help of neighbours, so collaborative misbehaviour 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 behaviour of the nodes, so there are more chances for false positives.

2.4.3. CORE

Pietro Michiardi and Refik Molva (2002) 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 behaviour to be rewarded and bad behaviour to be punished. This method gives more importance to the past behaviour and hence tolerable to sporadically bad behaviour, e.g. battery failure. But the assumption that past behaviour to be indicative of the future behaviour may make the nodes to build up credit and then start behaving selfishly.

2.4.4. CONFIDANT

CONFIDANT was proposed by Buchegger et al., (2002). Here evidence from direct experiences and recommendations is collected. Trust relationships are established between nodes based on collected evidence and trust decisions are made based on these 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 neighbour after forwarding a packet to the neighbour. It then reports to the reputation system only if the collected evidence represents a malicious behaviour. Reputation system changes the rating for a node if the evidence collected for a node’s malicious behaviour 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. But it cannot handle some of the routing attacks by malicious nodes.

2.4.5 Reputation-based System for Encouraging the Cooperation of Nodes

Tiranuch Anantvalee and Jie Wu (2007) in their paper, introduces, besides cooperative nodes and selfish nodes, a new type of node which is a suspicious node. These suspicious nodes will be further investigated and if they tend to behave selfishly, some actions against them would be taken, like we do to selfish nodes to encourage them to be cooperative. They introduce the use of a state model to decide what to do or respond to nodes in each state. In addition to a timing period for controlling when the reputation should be updated, a timeout for each state is introduced.

The propagation of reputation values may be used to attack a legitimate node by accusing it as a selfish node. To alleviate this problem, only second-hand information that is compatible with a node’s own information will be used. Even if the information is used, it should only slightly influence the resulting reputation values. Once a node is determined as selfish, the punishment is carried out to penalize the selfish node.

To receive a better service, routing through selfish nodes is avoided by selecting a route without them. In addition to obtain only routes free of them, selfish nodes may be added to an “avoid list; an additional field, when sending a Route Request. When using reputation, most of the existing
systems use a binary threshold to determine whether a node is well-behaving or selfish. In this system, they use two thresholds to categorize nodes into three categories. If the reputation value is above the first threshold or below the second threshold, it is cooperative or selfish, respectively. The third type of node is a suspicious node, whose reputation value falls between two thresholds. It cannot be determined right away whether they are cooperative or selfish, so they should be further investigated. If they tend to behave selfishly, some action is taken against these suspicious nodes to encourage them to be cooperative. In addition to using two thresholds, use of a state model to determine the states of other nodes and to decide what should be done in each state is introduced. What state a node is put into depends on its reputation rating and its previous state. Moreover, the use of a timing period to control when the reputation should be updated and to be used as a timeout for each state is introduced. But this paper does not deal with malicious nodes.

2.4.6 Cooperative On-demand Secure Route Protocol

Cooperative On-demand Secure Route (COSR) proposed by FeiWang (2010), is a novel secure source route protocol, against malicious and selfish behaviours. COSR measures node reputation (NR) and route reputation (RR) by contribution, Capability of Forwarding (CoF) and recommendation upon Dynamic Source Route (DSR) and uses RR to balance load to avoid hot point. Furthermore, COSR defines path collection algorithm by NR to enhance efficiency of protocol. This paper addresses the problems like DoS attack, Black-hole attack, Rushing attack, Wormhole attack and also selfish nodes. In the COSR, node’s reputation depends on the information from Physical layer, Media Access Control (MAC) layer, and Network layer, and it can be computed by node’s CoF, history action, and recommendation. COSR can be divided into the following modules; monitor, statistics, reputation model, reputation protocol, and routing protocol.
(i) Monitor: This part includes three modules: neighbour monitor, data relay monitor, and CoF monitor. Neighbour monitor works with MAC layer. It is used to monitor neighbours in its radio range and maintain neighbour list. Data relay monitor is placed in the network layer. It requires MAC layer working in promiscuous mode, so that it could check whether the next hop had transmitted its packets. CoF would collect information about capability of forwarding from physical layer and MAC layer, and it includes node’s bandwidth, interface state, mobility status, and power.

(ii) Statistics: This module is responsible for providing statistical data about neighbours’ history behaviour. These data include the number of requested and forwarded protocol messages, and data packets.

(iii) Reputation model: This is the core module of COSR. It is used to evaluate node’s reputation and integrate route reputation relying on the data from Monitor and Statistics.

(iv) Reputation Protocol: This part defines reputation discovery in the MANET. Reputation Protocol clings with routing protocol and uses routing protocol to pigback reputation control message and data.

(v) Routing Protocol: It is an extension of DSR by reputation model. It uses NR and RR to choose the best route path rather than path length. Further, COSR provides a secure mechanism of path collection in route discovery.

The CoF is the new concept introduced in this paper. CoF denotes the capability of forwarding packets of a certain node. Simply, they use the remained power, bandwidth, and mobility state to evaluate it. In the CoF, remained power and bandwidth are mandatory; however, mobility state
is optional. Only when the node supports Global Positioning System (GPS), it should provide mobility state and its velocity. As the information of CoF is provided by its owner, malicious node might cheat others by false data.

To avoid the emergence of such malicious behaviour, COSR takes the following strategies.

1) Discounting. COSR uses node’s reputation to discount those providing CoF data.
2) Punishment. Once COSR finds that any node provided a false CoF, it will punish such node through reducing its reputation level.

But the authors have not clearly specified how COSR will decide whether the advertised information is false or not.

2.5. ACKNOWLEDGEMENT BASED APPROACHES

This section explains approaches which are based on acknowledgements from the destinations or intermediate nodes.

2.5.1 Acknowledgement based approach for detecting routing misbehaviour

2ACK is proposed by Kejun Liu et al., (2006). This approach considers only packet forwarding misbehaviour. When a node forwards data packet successfully 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 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 the data
packet, but it simply sends 2ACK which act as an acknowledgement for the 2 hop neighbour.

Sundararajan, T.V.P and Shanmugam.A (2009) proposed a method which also follows 2ACK scheme but works on AODV 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.

2.5.2 TWOACK Scheme

Balakrishnan et al., (2005) proposed a TWOACK scheme which can be implemented as an add-on to any source routing protocol. Instead of detecting particular misbehaving node, TWOACK scheme detects misbehaving link and then seeks to alleviate the problem of routing misbehaviour by notifying the routing protocol to avoid them in future routes. It is done by sending back a TWOACK packet on successful reception of every data packet, which is assigned a fixed route of two hops in the direction opposite to that of data packets. Basic drawback of this scheme is; it cannot distinguish exactly which particular node is a misbehaving node. Sometime well behaving nodes became part of misbehaving link and therefore cannot be further used in the network. Thus a lot of well-behaved node may be avoided by network which results in losing of well-behaved routes.

2.5.3 Secure 2ACK Routing Protocol in Mobile Ad-hoc Networks

Vijaya (2008) proposed another acknowledgement based scheme similar to TWOACK scheme, which is also integrated on top of any source routing protocols. This scheme detects the misbehaving link, eliminate it and choose the other path for transmitting the data. The main idea is to send 2ACK
packet which is assigned a fixed route of two hops back in the opposite direction of the data traffic route and to reduce the additional routing overhead, a fraction of the data packets will be acknowledged via a 2ACK packet. This fraction is termed as Rack and by varying the Rack, overhead due to 2ACK packets can be dynamically tuned. This scheme also consists of multicasting method by which sender can broadcast information of misbehaving nodes so that other nodes can avoid path containing misbehaving nodes and take another path for the data transmission. Although routing overhead caused by transmission of acknowledgement packets is minimized but this scheme also suffers to detect the particular misbehaving node.

2.5.4 Co-operative Approach to Detect Misbehaving Nodes in MANET

Usha and Radha (2009) proposed extension to the TWOACK scheme, in which each node must send back a normal ACK to its immediate source node after receipt of any kind of packet. This scheme requires an end to end ACK packet (i.e. NACK) to be sent between the source and the destination. On receipt of the data packets sent by the source, destination responds with a NACK packet. The NACK would reach the source from the destination with the help of the path, which is found in the actual message packet, delivered to the destination. If a node is found to be misbehaving in the pre calculated path, the intermediate nodes are free to divert the NACK packet through alternative paths and this path will be stored in the NACK packet along with the older path, which is extracted from the original message. On receipt of the NACK packet, the source node compares the two paths that are in the NACK packet. If variation is found, then the node in the source to destination path, from where the path varies in the destination to source path is isolated and that particular node is marked as a potential misbehaving node by the source node otherwise source node concludes no potential misbehaving nodes in the path. Possible drawback includes lot of routing overhead because
of ACK and NACK packets. Also due to nodes mobility probability of NACK packet reaching to source becomes smaller with the large number of intermediate nodes between source and destination.

2.5.5 Adding Security against Packet Dropping Attack in MANET

Zeshan et al., (2008) proposed a two-fold approach for detection and isolation of nodes that drops data packets. First approach attempts to detect the misbehaviour of nodes and will identify the malicious activity in network. It is done by sending an ACK packet by each intermediate node to its source node for confirming the successful reception of data packets. If the source node does not get ACK packet by intermediate nodes then source node send again its packet for destination after a specific time. If same activity was observed again then source node broadcast a packet to declare the malicious activity in the network. Other approach identifies exactly which intermediate node is doing malicious activity. It is done by monitoring the intermediate nodes of active route by the nodes near to active path which lies in their transmission range and by the nodes which are on the active route. Since monitoring nodes are in promiscuous mode and are in the transmission range of intermediate nodes of active route, they can receive all the packets sent along the active route. Monitoring nodes count the number of packet coming into and going out of the nodes of active route.

Each monitoring node maintain a list of sent and dropped packets and when number of dropped packets by a particular node exceeds certain threshold, the monitoring node in that range declares that node as misbehaving node and broadcast this information. Upon receiving broadcast packet all neighbouring nodes will cancel their transmission to that particular node and enter it into the list of misbehaving nodes. Main disadvantage of this scheme includes the overhead due to transmissions of acknowledgement
packets by every intermediate node to the source and working of all nodes in promiscuous mode. Another disadvantage is, there is a possibility that a misbehaving monitoring node may raise false alarm message about a well behaving node causing Denial of Service (Dos) attack.

2.5.6 Improved Acknowledgement Based Scheme in MANET

Aishwarya Sagar et al., (2010) proposed an improved approach over the existing ACK schemes. Here as soon as the active route is found, all nodes of active route are logically grouped into N sets (i.e. S₁, S₂,...,Sₙ) where N=n/3 (n is number of nodes on active route) such that set S₁ contains first three consecutive node, set S₂ contains next three consecutive nodes and so on. By forming a logical grouping among the nodes in the active route, the authors claim that they can reduce the routing overhead compared to other ACK based approaches. But this approach also assumes that the malicious nodes will not drop acknowledgement packets.

2.6 MOBILE AGENT BASED APPROACHES

This section explains some of the approaches which uses mobile agent for security in mobile ad-hoc networks. Most of the methods are using the mobile agents only for collecting the routing information.

2.6.1 Multi Agent Routing Protocol

Romit Roy Choudhury et al., (2004) devised an agent based framework with its associated protocols and mechanisms. The agents in the framework move from one node to another, giving and taking relevant information, with the primary objective of making all nodes in the system, topology-aware. This topology awareness is used in the context of establishing and maintaining a communication link between two nodes. The mobile agent
is used just to update the routing information in all the nodes it visits. It has nothing to do with access control.

2.6.2 A Mobile Agent based Routing Protocol

Shivanajay Marwaha et al., (2002) proposed a novel routing scheme for mobile ad-hoc networks (MANETs), which combines the on-demand routing capability of Ad-hoc On-Demand Distance Vector (AODV) routing protocol with a distributed topology discovery mechanism using ant-like mobile agents. This protocol tries to overcome the shortcomings of existing routing protocols by using ant based routing and AODV, combining them to develop a hybrid routing scheme. Ants in network routing applications are simple agents embodying intelligence and moving around in the network from one node to the other and updating the routing tables of the nodes they visit with what they have learned in their traversal so far. Routing ants keep a history of the nodes previously visited by them. When an ant arrives at a node it uses the information in its history for updating the routing table at that node with the best routes it has for the other nodes in the network. All the nodes in the network rely on the ants for providing them the routing information, as they themselves do not run any program for finding routes. The ant-based routing algorithm implemented in this paper does not consider any communication among the ants. Each ant works independently. But this protocol does not deal with malicious behaviour of nodes.

2.6.3. Agent based Ad-hoc On-demand Distance Vector

M.Lakshmi et al., (2006) proposed an agent based power aware AODV which uses distributed discovery mechanism using mobile agents. These agents perform important tests and generate best power aware route through a network. This protocol uses both static and mobile agents. Agent
based AODV utilizes agents working independently and providing routes to the nodes. Each node will have an agent server running that provides basic functionalities for the static and mobile agents such as security, communication and migration. The use of mobile agents with AODV increases the node connectivity and reduces the route discoveries thus reducing the overall power consumption. This approach also does not handle any misbehaving nodes.

2.6.4 Mobile Agent Based Detection of Selfish Node in MANET

Debdutta Barman Roy et al., (2011) proposed a mobile agent based method for detecting selfish nodes in MANETs. Here a new Intrusion Detection System (IDS) is proposed based on Mobile Agents. The approach uses a set of Mobile Agent (MA) that can move from one node to another node within a network. This as a whole reduces network bandwidth consumption by moving the computation for data analysis to the location of the intrusion. A source node sends a Route Request using normal DSR and finds a route to the destination. It transmits data through the discovered path and waits for acknowledgement from the destination. If it does not receive acknowledgement within the stipulated time, then the source nodes generates a mobile agent and sends it through the route. The mobile agent on reaching the next hops in the source list calculates the packet delivery ratio and finds out a node which was selfish. On identifying the selfish behaviour of a mobile node, it sends a report regarding the malicious activity of the node to the source node. Otherwise it moves to the next hop and so on.

Debdutta Barman Roy et al., (2012) also proposed a method for identifying black hole attacks using the mobile agents. But it is not clear how a mobile agent knows the number of packets forwarded and number of packets received at any node to calculate the packet delivery ratio and to identify it as
a malicious node. The advantage of this method is, it allows the mobile agent to do the calculations instead of putting burden on the host.

2.7 SUMMARY

Thus this section has explained the existing approaches for access control in mobile ad-hoc networks. Credit based or incentive based approaches enforce cooperation among mobile nodes by giving incentives or credits for well behaving. One problem with some credit based systems is that they need a tamper proof hardware module for proper functioning of the method. Ticket based approaches provide tickets by using cryptographic methods and only a node with valid ticket is allowed to access the network. But the overhead involved in sharing the keys is becoming a problem with these types of approaches. Secure routing protocols modify the existing protocols in MANETs or develop new routing protocols for identifying and isolating misbehaving nodes from the network. Each secure protocol has its own advantages and disadvantages.

A comparative study of these protocols are also done by Raju Barskar (2011), Dipali Koshti and Supriya Kamoji (2011), Sangheethaa Sukumaran et al., (2009) etc. Reputation based systems distinguish between the well behaving and misbehaving nodes by calculating the reputation value which depends on the past behaviour of the nodes. Each approach calculates the reputation using some equations and consists of different components like monitoring agent, reputation calculator, and route selecting agent and so on. Reputation based systems are normally implemented over any of the existing routing protocols. Acknowledgement based approaches are using the concept of feedback mechanism for positive or negative reception at the destination. These acknowledgements are sent either by destination alone, or by the intermediate nodes or by the set of nodes in between. Anyhow these
acknowledgement based approaches are not dealing with the loss of acknowledgements, or the authenticity of acknowledgements. Mobile agent based approaches are recently evolved. Mobile agents are used for collecting the routes from other nodes in the network. They are also used to update the routing tables of other nodes. But no approach have used mobile agent for finding a trustworthy route. This thesis uses the mobile agents for finding the trustworthy route out of the available routes which is explained in the later chapters.