A Security Model based on Reputation and Collaboration through Route-Request in Mobile Ad Hoc Networks

Anjali Anand[1], Rinkle Rani[2] and Himanshu Aggarwal[1]
[1] Department of Computer Engineering, Punjabi University,
Patiala-147002, India
[e-mail: anjalianand_87@yahoo.in, himanshu.pup@gmail.com]
[2]Computer Science and Engineering Department, Thapar University,
Patiala-147004, India
[e-mail: raggarwal@thapar.edu]
*Corresponding author: Anjali Anand

Received May 10, 2015; revised August 11, 2015; accepted September 4, 2015;
published November 30, 2015

Abstract

A Mobile Ad hoc Network (MANET) consists of mobile nodes which co-operate to forward each other's packets without the presence of any centralized authority. Due to this lack of centralized monitoring authority, MANETs have become vulnerable to various kinds of routing misbehaviour. Sometimes, nodes exhibit non-cooperating behaviour for conserving their own resources and exploiting others' by relaying their traffic. A node may even drop packets of other nodes in the guise of forwarding them. This paper proposes an efficient Reputation and Collaboration technique through route-request for handling such misbehaving nodes. It lays emphasis not only on direct observation but also considers the opinion of other nodes about misbehaving nodes in the network. Unlike existing schemes which generate separate messages for spreading second-hand information in the network, nodes purvey their opinion through route-request packet. Simulation studies reveal that the proposed scheme significantly improves the network performance by efficiently handling the misbehaving nodes in the network.

Keywords: reputation based scheme, MANET security, misbehaving node, misbehaving node detection, routing misbehaviour.

http://dx.doi.org/10.3837/tiis.2015.11.025 **ISSN : 1976-7277**

1. Introduction

A mobile ad hoc network (MANET) is a collection of mobile nodes or routers which communicate through wireless links forming an arbitrary topology. MANETs are different from conventional networks in the sense that they are dynamic in nature having no fixed infrastructure. Ad hoc Networks provide communication in the absence of a fixed infrastructure without a central administrating authority, making them suitable for various applications such as rescue operations, disaster recovery, tactical operations, environmental monitoring, mobile conferences, etc. There are certain inherent features of MANETs. Nodes in the network move independently in an unrestricted geographical topology. Each node can freely join or leave the network as there is no specific entry or exit point. Nodes have limited battery life, memory and CPU capacity. Owing to restricted transmission range of the devices, nodes in the network rely on their neighbours to route their traffic to the destination. Dynamic Source Routing (DSR) [1] is an on-demand routing protocol for MANETs. Besides DSR, other popular routing protocols for MANETs are Ad hoc On-demand Distance Vector (AODV) and Optimized Link State Routing (OLSR) protocol. The details and performance evaluation of AODV and OLSR can be found in [2]. Features such as rapid deployment, flexibility and adaptability make MANETs a very promising technology.

A mobile ad hoc network can either be closed or open. In a closed MANET, nodes work together towards achieving a common objective as in military and rescue operations. Whereas, in an open MANET, individual nodes having different aims co-operate to provide global connectivity by sharing resources. But, there exists a trade-off between co-operation and consumption of resources as providing network services such as routing and forwarding may consume a lot of resources like local CPU time and memory, battery power and network bandwidth. A node may wish to conserve such resources while exploiting others by using their services. A node exhibiting such behaviour is termed as misbehaving node. In MANETs, presence of misbehaving nodes can severely deteriorate the network performance as they can refuse to participate in network functions. Hence, it is essential to devise a mechanism which is efficient and accurate in detecting routing misbehaviour and alleviates its effect. This paper presents a Reputation and Collaboration technique which efficiently handles routing misbehaviour.

Several approaches have been given in the literature discussing the problem of routing [3][4][5][6][7][8][9]. These approaches can be widely categorized as: credit-based or incentive-based schemes [10][11][12], acknowledgement schemes [13][14][15] and reputation-based schemes [16][17][18][19][20][21][22].

CONFIDANT [16] is a reputation based scheme for handling misbehaviour in MANETs. It uses a combination of local and second-hand information (in the form of alarm messages) for maintaining the reputation of nodes. It uses Bayesian Estimation for calculating trust relationship and reputation of nodes which makes the misbehaviour detection process complex. The disadvantage of CONFIDANT scheme is that it can fall prey to false accusations as nodes tend to lie and liar nodes may conspire against other nodes.

CORE [17] provides a collaborative monitoring and reputation technique for node co-operation. The disadvantage of this scheme is that it assumes that all nodes calculate the reputation in the same way and assign same weight to all the functions which may be inappropriate in heterogeneous networks. Devices may choose to set different importance levels for various functions based upon their CPU and battery usage etc.

OCEAN [18] falls into the category of local reputation-based scheme as it solely depends upon the direct observation of neighbours for maintaining reputation information. It modifies the rating for the neighbour node based on its forwarding behaviour. When the rating of a node reaches minimum threshold, it is added to the 'Faulty List' which is a list of misbehaving nodes. OCEAN introduced the concept of 'avoid list' for handling misbehaviour. It is a list of nodes which the sender wishes to avoid in its future routes. The sender sends the 'avoid list' along with the route request packet. The nodes in the network discover a path avoiding the nodes listed in the 'Avoid List'. In certain cases, depending upon direct observation solely may be inappropriate.

Locally Aware Reputation System (LARS) [19] is a direct observation based reputation system for detecting and handling selfish nodes. A node's misbehaviour is communicated in the network in the form of WARNING message. These messages are signed using threshold cryptography technique to avoid false accusations.

In [20], Al-Karaki and Kamal have proposed a technique for node co-operation by combining reputation-based and virtual currency based schemes. Every node records the behaviour of its neighbours, in the form of a table. This table is communicated to all other nodes in the network. One of the drawbacks of this scheme is its requirement of electing a ClusterHead which is responsible for communication between different clusters. The election process is quite tedious as managing the tasks of ClusterHead is resource-intensive and nodes may refrain themselves from being elected.

LMRSA [21] is a reputation mechanism that does not consider second-hand information and is based only on direct observation. Once detected, misbehaving nodes are isolated from the network. LMRSA generates explicit alert messages to inform source node about misbehaving nodes in the route. Source node, then prunes the route containing the misbehaving nodes from its route cache.

In [22], a trust and reputation based mechanism has been proposed to detect and handle selfish and malicious nodes. It uses direct observation and on-demand second-hand information (recommendation messages) for calculating aggregate reputation and trust value for the nodes. A notification mechanism is used to inform the source node about a misbehaving node which then uses explicit message (REREQ and REREP) for checking the misbehaviour of the accused node to prevent false accusation. This technique is referred as TBRM in **Table 1.**

The proposed scheme is a Reputation and Collaboration technique which does not solely depend upon direct observation of nodes, as in case of OCEAN [18]. It uses a combination of first-hand information and opinion of other nodes for dealing with routing misbehaviour. Unlike CONFIDANT [16] and LARS [19], it does not spread second-hand information in the form of alarm messages. Nodes collaborate by purveying their opinion about misbehaviour of other nodes through Route-Request packet instead of generating separate alarm messages. **Table 1** shows the comparison of the Proposed Scheme with the existing reputation-based schemes.

Table 1. Comparison of the proposed scheme with existing reputation-based schemes

FEATURES	LARS [19]	CORE [17]	CONFIDANT [16]	OCEAN [18]	TBRM [22]	LMRSA [21]	PROPOSED SCHEME
Direct Observation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Second-Hand Information	No	Yes	Yes	No	Yes	No	Yes
Avoids Alarm/Explicit Messages	No	Yes	No	Yes	No	No	Yes
Handles False Accusation	does not exist	No	No	does not exist	Yes	No	Yes
Avoids misbehaving node in route discovery	Yes	No	No	Yes	Yes	Yes	Yes
Punishment	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The next section gives description of routing misbehaviour problem. Section 3 includes the detailed working along with the algorithm for the proposed scheme. Section 4 entails the simulation environment and performance analysis of the proposed scheme. The last section presents the conclusion and future scope of the proposed work.

2. Routing Misbehaviour Problem

This section discusses the problem of routing misbehaviour along with the threats posed by it to the communication in mobile ad hoc networks.

2.1 Modelling Routing Misbehaviour

The proposed scheme is capable of detecting and handling the following types of misbehaviour:

- *Packet Drop*: A node actively participates in the route formation but later drops packets of other nodes in the guise of forwarding them. Hence, misleading other nodes into unsuccessfully sending their traffic through it. Such nodes act as sink; posing threat to resource availability in the network.

- *Non-participation*: A node may not respond to route request for preserving its resources but may exploit others by sending its own traffic through the network. It enjoys network services but refuses to forward the packets of other nodes. Its participation is limited to just sending and receiving its own packets. Such nodes deny services to other nodes causing serious threat to availability of network services.
- *Tampering of data packets*: A node participates in route discovery process but once the route is established and packets are being transmitted, it may tamper with the data while forwarding it. This kind of misbehaviour is a threat to integrity of data shared in the network.
- *Partial data forwarding*: A node may stop forwarding packets after transferring a few of them and try to cheat the monitoring system. Such nodes reduce the availability of resources to other nodes in the network.
- *Unintentional Packet Drop*: A node willing to participate in network operations may fail to do so due to failure. It has no intention of dropping packets of other nodes but is unable to relay them due to some failure in the system, such as transient link failure, etc. As the node is unable to forward the packet within a specified time limit, the packet is declared as being dropped.

The next sub-section shows the effect of such misbehaving nodes on the performance of the network. It includes a probability model which highlights how the existence of misbehaving nodes can hamper the successful transmission of data packets from the source to destination.

2.2 Misbehaviour Probability Model

This sub-section demonstrates how the presence of misbehaving nodes hampers the network performance.

Routes in the network between the source and the destination have been studied. A route is considered as misbehaving if out of all the nodes in the route at least one node is found to be misbehaving. A transmission is considered successful if the route from source to destination does not contain any misbehaving nodes. A route consists of average number of 'δ' nodes from the source to the destination. Therefore, the number of intermediate nodes in a route is 'δ- 2' i.e., except the source and the destination. The probability that a node may misbehave is given by $Prob_m$. Therefore, the probability of having at least one misbehaving node in the route is given by:

$$\mathrm{Prob}_{\mathrm{r}} = 1 - (1 - \mathrm{Prob}_{\mathrm{m}})^{\delta-2}$$

Hence, the probability that a packet will be successfully delivered to the destination is given by:

$$\mathrm{Prob}_{s} = 1 - \mathrm{Prob}_{r}$$

or

$$\mathrm{Prob}_{\mathrm{s}} = (1 - \mathrm{Prob}_{\mathrm{m}})^{\delta-2}$$

Table 2 illustrates the effect on probability of successful transmission of increasing probability of misbehaving nodes in the network by keeping the average number of nodes in a route ‘δ‘ as constant. From **Table 2**, it can be concluded that $Prob_r$ increases with an increase in $Prob_m$. It can be seen that almost half of the routes may become futile when the probability of misbehaving nodes $Prob_m$ is 0.2. The probability of successful transmission, $Prob_s$ also decreases with an increase in the value of $Prob_m$. It has been noticed that the probability of successfully delivering a packet $Prob_s$ is around 0.51 when the probability of misbehaving node $Prob_m$ is 0.2.

Table 2. Probability of successful transmission for varying probability of misbehaving nodes

δ=5		
$Prob_m$	$Prob_r$	$Prob_s$
0.1	0.27	0.73
0.2	0.49	0.51
0.3	0.34	0.66

Table 3 displays the probability of successful transmission by increasing ‘δ’ i.e. the average number of nodes in a route, and keeping $Prob_m$ as constant. As the size of network increases, value of ‘δ’ also increases. The probability of successful transmission $Prob_s$ is inversely proportional to ‘δ’. Therefore, the probability of successful transmission $Prob_s$ decreases with an increase in the value of ‘δ.

Table 3. Probability of successful transmission for varying number of nodes in a route

$Prob_m$ = 0.2		
δ	$Prob_r$	$Prob_s$
4	0.36	0.64
6	0.59	0.41
8	0.73	0.27

From the above discussion it can be concluded that a high probability of misbehaving nodes can have a profound impact on the network performance. As the network size increases, the situation further worsens. Hence, it is indispensable to provide an efficient mechanism for detecting and alleviating routing misbehaviour in MANETs which is the basis for this research work.

3. Detecting and handling Routing Misbehaviour

The proposed scheme provides an efficient reputation and collaboration technique through route request for handling routing misbehaviour. Neither does it merely depend upon direct observation of nodes, as in case of OCEAN [18] and LARS [19], nor does it aggressively spread second-hand information about the behaviour of nodes in the network, like CONFIDANT [16]. It lays more emphasis on direct observation but does not completely ignore the opinion of other nodes in the network. It creates a concoction of both direct observation and second-hand reputation in a way which is beneficial for the system. Rather than spreading alarm messages, nodes collaborate to inform other nodes about misbehaving nodes through route request. The proposed scheme works with Dynamic Source Routing (DSR) Protocol to provide security against misbehaving nodes in mobile ad hoc networks.

The following assumptions are made for the network considered in this paper.

- Nodes are distributed uniformly within the network area.
- Traffic is randomly distributed among nodes.
- Misbehaving probability of a node is independent of the misbehaving probability of any other node in the network.
- Source and destination nodes are randomly selected for various transmissions.

3.1 Lists used in the scheme

The proposed scheme is using the following lists for handling routing misbehaviour:

- *Misbehaving Nodes' List* L_M: Each node maintains a list of nodes it considers as misbehaving. The nodes are added into the list as soon as their misbehaviour is detected.
- *Ancillary List* L_A: Each node creates an Ancillary List L_A from nodes present in its Misbehaving Nodes' List L_M. It consists of two fields: *h_id* and *m_id*. The *h_id* field contains the ID of the accusing node and the *m_id* field contains the ID of the node which the current node considers to be misbehaving. If a node considers more than one node as misbehaving then separate entry will be made in the list for each misbehaving node.
- *Global Misbehaving Frequency List* L_{GMF}: Global Misbehaving Frequency (GMF) List L_{GMF} keeps the count of the number of nodes which consider a particular node as misbehaving. The GMF List consists of two fields: *m_id* and *frequency*. The *m_id* field contains the ID of the node which is considered to be misbehaving and the *frequency* field contains the number of nodes in the network which consider the corresponding node as misbehaving.

3.2 Promiscuous Watch

This module monitors the behaviour of the neighbouring nodes. It is used for tracking the misbehaving nodes in the network. It consists of two sub-modules: Event Register and Rating Calculator. When a packet is forwarded, the Event Register sub-module saves the checksum of the packet in a buffer before sending it. After sending the packet, it monitors the wireless channel in promiscuous mode to check whether the neighbour is forwarding it or not. If the neighbour fails to forward the packet in time T_{PW}, a negative event is registered against the neighbour node and the packet checksum is removed from its buffer. Whereas, if the sender node overhears an attempt to forward the packets by the neighbour node, it compares checksum saved in its buffer with that of the forwarded packet. If a match is found, a positive event is registered and the checksum is removed from its buffer. If the checksum does not match, the packet is treated as not been forwarded. It then communicates these events to the Rating Calculator.

Rating Calculator maintains rating R for its neighbour nodes. The default value of rating is set to $R_{Neutral}$ which is incremented by an amount R_{inc} on receiving a positive event and decremented by amount R_{dec} on receiving a negative event from the Event Register module.

$$R = \begin{cases} R + R_{inc}, if \quad positive\ event \\ R - R_{dec}, if \quad negative\ event \end{cases}$$

The Promiscuous Watch module monitors not only the behaviour of the neighbouring nodes but also verifies that the packet is not modified or tampered before being forwarded by the neighbour node. When it hears a packet is forwarded by the neighbour node, it matches the forwarded packet with the saved checksum. If the neighbour node modifies the contents of the packet, the checksum will not match and the packet will be treated as not being forwarded. This module also keeps a check on partial data forwarding by neighbour nodes. Neighbour node, in order to save its resources, may stop forwarding the packets after transmitting a fraction of packets. Promiscuous Watch module effectively handles such misbehaviour as each packet is monitored after sending it to the neighbour node.

3.3 Handling Packet Drop Misbehaviour

Every node maintains a List L_M for storing the nodes which it considers to be misbehaving. When a node's rating R is less than Th_m i.e., $R <= Th_m$, it is considered as misbehaving and is added to the List L_M.

3.3.1 Using Ancillary List

A variable-length field is appended to DSR Route-Request (RREQ) packet for handling packet drop misbehaviour. Each node appends its own Ancillary list L_A to this field. When the source node generates its DSR Route-Request Packet, it appends its List L_A to it. The source node then broadcasts this packet on the wireless medium. Upon receiving a DSR Route-Request Packet, intermediate node appends its own List L_A to the packet and re-broadcasts it.

The collaboration among the sender node, intermediate nodes and the receiver node using Ancillary List L_A in route request packet is discussed in the following:

- **Sender Node**

At the sender, the Promiscuous Watch component maintains the rating R of each neighbouring node and the nodes which are found to be misbehaving are added into the Misbehaving Nodes' List L_M. Ancillary List L_A is created from the nodes in the List L_M and appended to the Route-Request (RREQ) packet generated. The complete packet is then broadcasted to other nodes. **Fig. 1** shows the actions performed by the sender node.

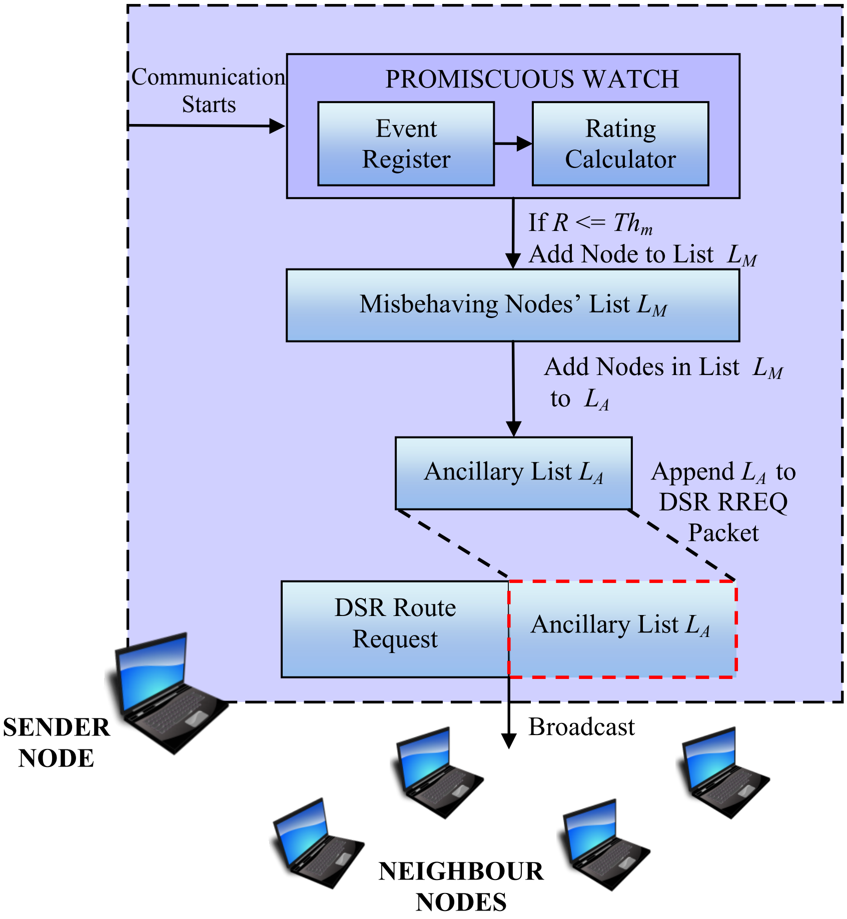

Fig. 1. Actions performed by the sender node

- **Intermediate Node**

When an intermediate node receives a RREQ packet, it first updates its GMF list L_{GMF} and then appends its own list L_A to the RREQ packet received and re-broadcasts it. For considering the opinion of other nodes along with handling false accusation two system parameters i.e., $R_{Dubious}$ and Th_f are employed. If a single node falsely accuses a well behaving node and advertises it as 'misbehaving' in its List L_A, its opinion would not be considered by other nodes in the network. The opinion of nodes is taken into consideration only if at least 'Th_f' nodes have the same opinion about a particular node in the network. During a transmission, if the number of nodes considering a node to be misbehaving is greater than or equal to Threshold Frequency Th_f i.e., *frequency* $>= Th_f$, then the rating R of the corresponding node is set to $R_{Dubious}$ so that its misbehaviour can be detected as soon as communication begins.

$$R = R_{Dubious}\ ,\ if\ \ frequency >= Th_f$$

$R_{Dubious}$ is the rating value given to the accused node. The value of $R_{Dubious}$ is taken as -30 which is kept close to Th_m (-40) to ensure the rapid addition of the node to List L_M on direct communication with it. If 'Th_f' or more nodes perform colluding attack by falsely accusing a node and advertise it as 'misbehaving', the proposed scheme does not directly add the accused node to L_M. It only sets its rating to $R_{Dubious}$ and the accused node still has a chance to improve its rating during direct communication with other nodes and prevent itself from being added to L_M of other nodes. These system parameters are flexible and can be adjusted according to network scenario and security requirements. **Fig. 2** illustrates the actions performed by an intermediate node on receiving a route request packet.

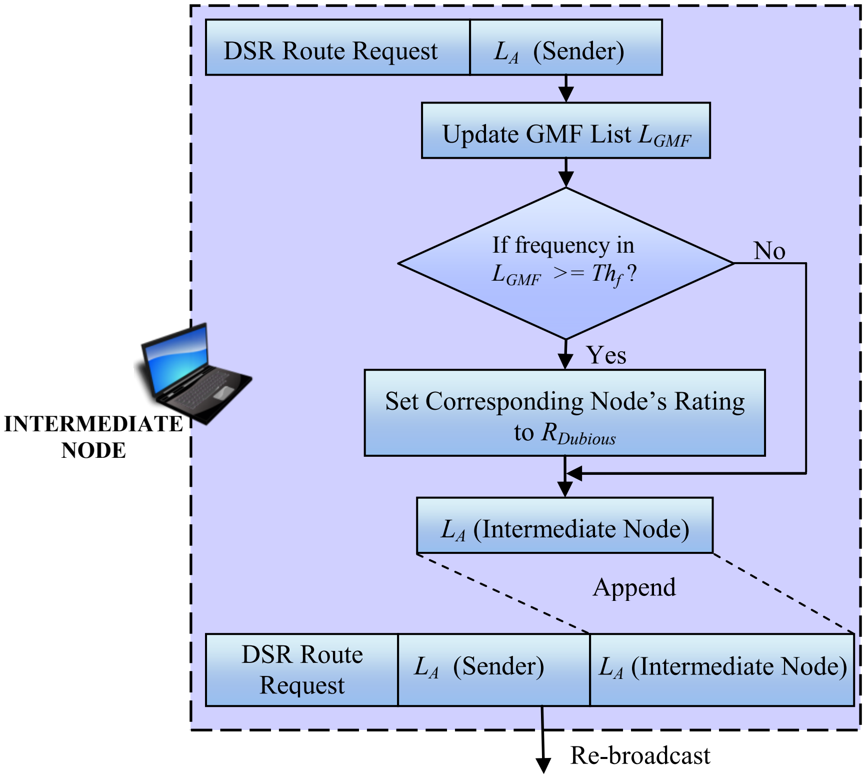
DSR Route Request
L_A (Sender)
Update GMF List L_{GMF}
If frequency in L_{GMF} >= Th_f?
No
Yes
Set Corresponding Node's Rating to $R_{Dubious}$
L_A (Intermediate Node)
Append
DSR Route Request
L_A (Sender)
L_A (Intermediate Node)
Re-broadcast
INTERMEDIATE NODE

Fig. 2. Actions performed by intermediate node

- **Receiver Node**

When the receiver receives a Route-Request packet, it extracts the List L_A in the packet and updates its GMF list L_{GMF} accordingly. It then checks the frequency of all nodes in list

L_{GMF}. If frequency of any node is found to be greater than or equal to Threshold Frequency Th_f i.e., *frequency* $>=$ Th_f, then its rating R is set to $R_{Dubious}$. Finally, the receiver generates a route reply which is sent back to the sender.

Algorithm 1 shows the handling of routing misbehaviour using Ancillary List L_A and how a node updates the *frequency* field in list L_{GMF} from the information contained in list L_A. It also illustrates how a node resets the rating R of a node to $R_{Dubious}$ when the value of *frequency* is greater than or equal to Threshold Frequency Th_f.

Algorithm 1: Using Ancillary List

```
while (End of List $L_A[h_{id}][m_{id}]$)
    if ((h_id != Examining Node's address) &&
     (m_id != Misbehaving Node's address)) then
        set h_id = Examining Node's address
        set m_id = Misbehaving Node's address
        set frequency =1
    end
    elseif (h_id = Examining Node's address) &&
      (m_id = Misbehaving Node's address)) then
        Return
    end
    else
        set frequency = frequency +1
        if (frequency >= $Th_f$) then
            set R= $R_{Dubious}$
    end
end
```

- **Route-Request Drop**

Upon receiving a RREQ packet, the intermediate node checks the Ancillary List L_A present in it. If the intersection of nodes in the List L_A and Route-Request route is non-void (i.e. a node considered as misbehaving is in the route), the Route-Request packet is dropped. Else the node appends its own List L_A to the RREQ packet and re-broadcasts it.

3.3.2 Traffic Rejection

The function of this module is to reject any kind of traffic from misbehaving nodes in the List L_M. The strategy of rejecting is adopted for disallowing misbehaving nodes to send their own traffic by disguising to relay it on some other node's behalf. When a neighbouring node sends a route request, the node checks its Misbehaving Nodes' List L_M.

If the requesting node is not present in its List L_M only then the route request is forwarded, else the route request is dropped.

Similarly, when a node receives a route reply from a neighbouring node, it first checks whether the replying node is in its Misbehaving Nodes' List L_M or not. If it does not exist in the List L_M, only then the reply is accepted else it is rejected.

3.4 Handling of Non-participation Misbehaviour

Each node maintains a counter, known as credit-count C_{Cr}, for each of its neighbour. A node gains credit on forwarding a packet of another node. A node loses credit with a node upon asking it to relay a packet. Upon receiving a request for forwarding by a neighbour, the node checks the credit-count C_{Cr} of that neighbour node. If the credit-count C_{Cr} is non-zero only then the Route-Request is forwarded, else it is dropped. Each node increments the credit-count C_{Cr} of each of its neighbour by a value I after a fixed interval of time called bank time-out T_{Cr} .

$$C_{Cr} = C_{Cr} + I, when\ T_{Cr}\ \exp ires$$

This is done to prevent the deadlock problem when nodes may not forward packets for each other due to zero credit-count.

3.5 Handling Misbehaviour due to failure

A node willing to forward packets of other nodes' may be unable to do so due to some failure such as transient link failure etc. Such a node can be very easily misjudged as a 'misbehaving node'.

This can be unfair to a node as it may be experiencing some kind of failure such as hardware failure, link failure, restarting the network interface etc. To handle such problems, a mechanism known as Reprieval Mechanism has been used. The Reprieval Mechanism allows nodes which were earlier considered as 'misbehaving' to become a part of the network again. In the absence of this technique, once a node is considered to be misbehaving it would not get any chance to demonstrate its good behaviour. Therefore, a timeout-based mechanism is adopted to remove a misbehaving node from L_M after a certain time period, known as the Reprieval Timeout T_{Rep}. But its rating R is not increased to $R_{Neutral}$ after removal from the List L_M to rapidly add it back to the List L_M in the event of continued misbehaviour.

4. Simulation and Performance Evaluation

This section contains the simulation environment and comparison of the proposed scheme in the presence of misbehaving nodes against other reputation based schemes and normal DSR protocol. It also includes the simulation results for evaluating the performance of the proposed scheme.

4.1Simulation Environment and Performance Metrics

The network simulator ns-2 (version 2.29) [23] has been used to run the simulations. Random Way-Point Mobility Model has been used for generating scenarios for mobility of nodes in the network. 20 simulations have been performed to achieve 95 percent confidence level and the average value for each data point has been plotted. **Table 4** lists the fixed and constant parameters used in the simulation.

Table 4. Fixed and constant parameters for simulation

Fixed Parameters	**Value**	**Constant Parameters**	**Value**
Area	1000 X 1000 m^2	R_{inc}	1
Number of Nodes	60 Nodes	R_{dec}	2
Radio Range	250 m	T_{PW}	1 sec
Speed	0 to 30 m/s	Th_m	- 40
Sending Capacity	2 Mbps	Th_f	3
Packet Size	512 B	$R_{Dubious}$	- 30
Traffic	CBR	T_{Rep}	30 sec
Simulation Time	900 sec	T_{Cr}	10 sec

For evaluating the performance of proposed scheme, the following metrics have been used:

- **Packet Delivery Ratio** is defined as the ratio of number of data packets received to the number of data packets sent.
- **Routing Overhead** is taken as the ratio of amount of control information generated to the amount of data transmission.
- **Throughput** is defined as the number of data packets correctly delivered to the destination in an observed duration of time.

4.2 Simulation Results

For performance evaluation, the proposed scheme is compared with various reputation based schemes like OCEAN, LARS, LMRSA and Normal-DSR Protocol. The simulation results obtained are as follows:

Fig. 3 illustrates the Packet Delivery Ratio of the proposed scheme and above mentioned algorithms by varying the probability of misbehaving nodes $Prob_m$. The proposed scheme outperforms other reputation based schemes in terms of packet delivery ratio when $Prob_m$ is less than equal to 0.5. But when $Prob_m$ goes above 0.5 the performance degrades immensely and there is no benefit of communicating in a network having such high probability of misbehaving nodes. Hence, such a network must be discarded.

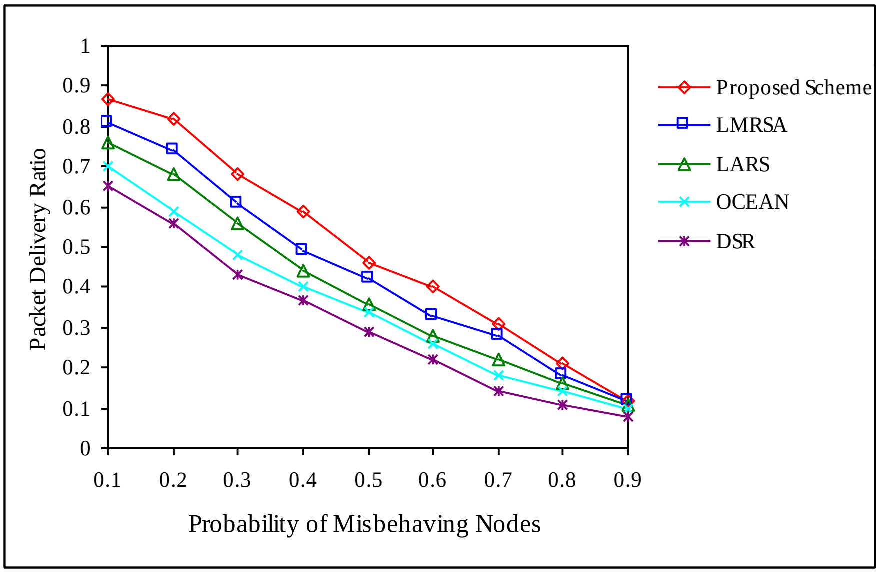

Fig. 3. Packet delivery ratio of the proposed scheme and other reputation based schemes

Fig. 4 highlights the throughput of misbehaving nodes in the proposed scheme and other reputation schemes along with Normal-DSR protocol. It can be observed that the proposed scheme has shown a steep decrease in the throughput of misbehaving nodes as $Prob_m$ increases. This is due to the rejection of traffic originating from misbehaving nodes by other nodes in the network.

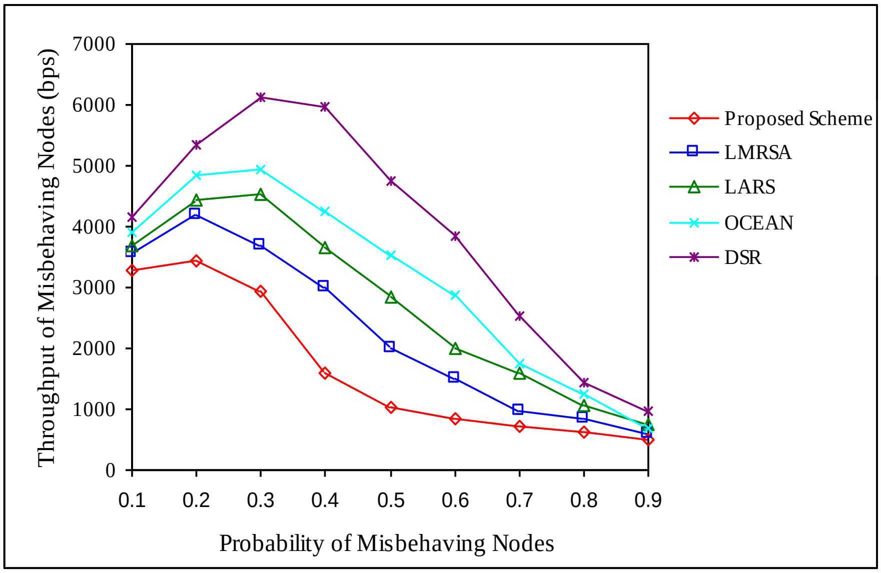

Fig. 4. Throughput of misbehaving nodes in the proposed scheme and other reputation schemes

The next best performing algorithm is LMRSA whereas; Normal-DSR protocol shows the worst performance due to absence of any kind of security mechanism.

Fig. 5 compares an important aspect of Routing Overhead in the proposed scheme and other reputation based algorithms. The amount of control information plays a vital role in the network. As the amount of control information increases, the time taken to establish routes also increases. The proposed scheme outperforms other reputation based algorithms in terms of Routing Overhead. As the proposed scheme uses RREQ packets (in the form of Ancillary List) to spread information about misbehaviour, the amount of control information generated is less as compared to other reputation based schemes like LARS and LMRSA. These schemes generate explicit messages to spread information about misbehaving nodes. With more misbehaving nodes, more messages are generated in the network thus; more routing overhead. The routing overhead of OCEAN is low as it does not spread information about misbehaviour.

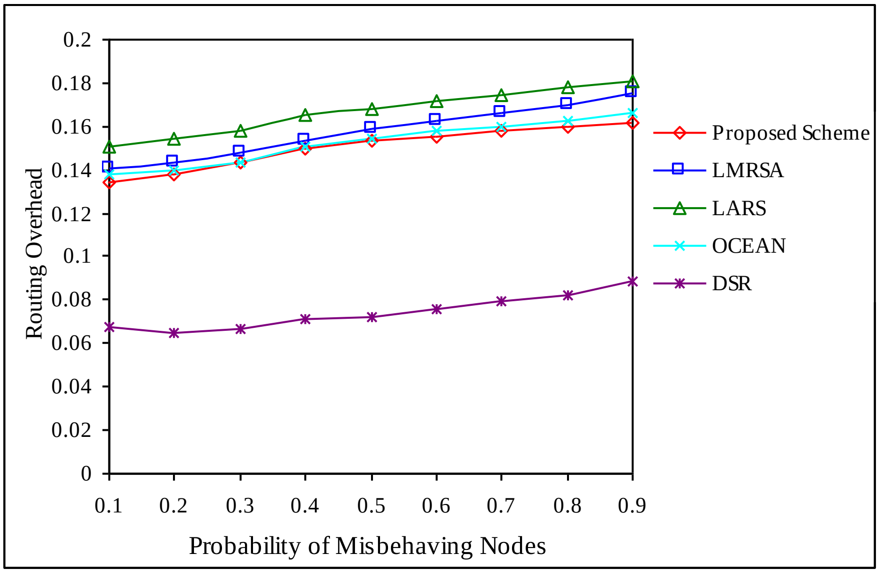

Fig. 5. Routing overhead in the proposed scheme and other reputation based schemes

Fig. 6 displays the fine tuning of Threshold Frequency Th_f in the proposed scheme. It is a crucial parameter for judging the system's performance. The Threshold Frequency Th_f in the proposed scheme varies from 1 to 5. The average number of packets dropped in the network is nearly the same when Threshold Frequency Th_f is set to 1, 2 and 3. But Th_f is selected as '3' because when Th_f is set to 1 and 2, the system becomes prone to false accusation by some misbehaving nodes**.** Nodes can perform a grouping attack by falsely accusing a well-behaving node and spread wrong information about it in the network. The probability of 3 nodes being involved in a grouping attack is less than the probability of 2 nodes performing a grouping attack. Even if 3 or more nodes perform a grouping attack on a node, the proposed scheme does not add the accused node directly to its Misbehaving Nodes' List L_M. Rather the accused node still has a chance to display its 'benevolence' and prevent itself from being added to the List L_M of other nodes.

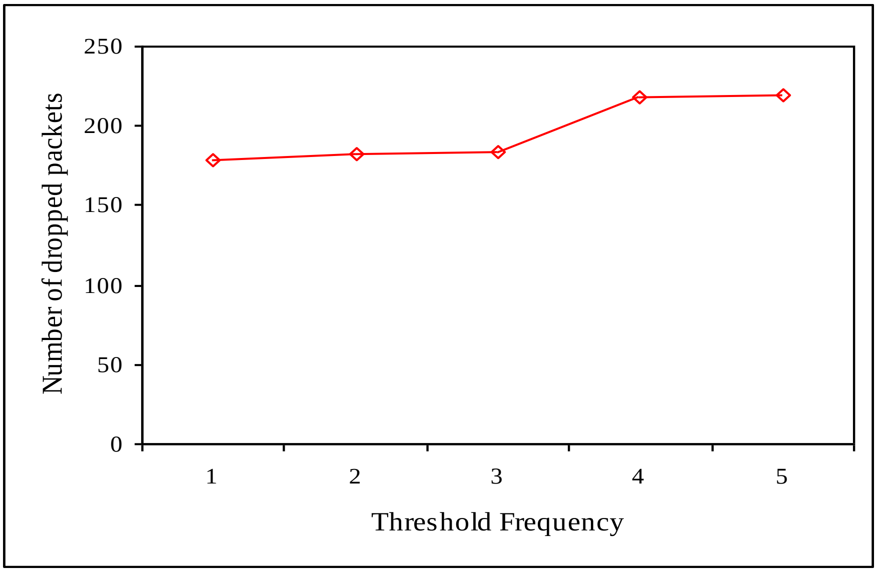

Fig. 6. Threshold frequency of proposed scheme

From simulation results, it is observed that the proposed scheme provides higher packet delivery ratio than other reputation based schemes. This is due to its ability to efficiently detect misbehaving nodes and provide secure routes for data transmission. Furthermore, the proposed scheme effectively suppresses the throughput of misbehaving nodes, enforcing them to concede their misbehaviour. The routing overhead in the proposed scheme is low as compared to other reputation based schemes as it uses route-request packets to spread information about misbehaviour, rather than generating explicit messages. In summary, it can be stated that the proposed scheme provides an efficient way of dealing with misbehaving nodes with increased packet delivery ratio and decreased routing overhead than other reputation based schemes.

5. Conclusion

This paper discusses the performance degradation in MANETs due to misbehaving nodes and presents an efficient reputation and collaboration technique through route-request for securing MANETs against such misbehaving nodes. The proposed scheme lays emphasis on direct observation but it does not completely ignore the opinion of other nodes. Consideration is given to opinion of other nodes in a way which is beneficial for the system, thereby, creating a blend of both the approaches. Rather than spreading alarm messages, nodes in the network collaborate to inform other nodes about misbehaving nodes through route-request using *Ancillary List.*

Detailed simulations have been conducted over the proposed scheme using ns-2 for performance evaluation of the proposed scheme. The results obtained from simulations indicate that the proposed scheme outperforms other reputation based schemes in the presence of misbehaving nodes. However, the scheme may be prone to tampering of Ancillary List by some misbehaving nodes. To overcome this problem, encryption of Ancillary List could be performed in future work. With further research, the proposed scheme is bound to show remarkable results in securing MANETs from the menace of misbehaving nodes.

References

[1] D.B. Johnson, D.A. Maltz and J. Broch, “DSR: The Dynamic Source Routing protocol for multi-hop wireless ad hoc networks,” In *Ad Hoc Networking, Chapter 5,* edited by C.E. Perkins ,Addison-Wesley, pp. 139-172, 2001.

[2] R. Malekian, A. Karadimce and A. H. Abdullah, "AODV and OLSR routing protocols in MANET,” in *Proc. of IEEE Thirty-third International Conference on Distributed Computing Systems Workshops*, pp. 286-289, 8-11 July, 2013. Article (CrossRef Link).

[3] A. Konig, D. Seither, R. Steinmetz and M. Hollick, “An analytical model of routing, misbehavior and countermeasures in mobile ad hoc networks”, in *Proc. of IEEE Global Telecommunications Conference*, pp. 1-6, 30 November- 4 December, 2009. Article (CrossRef Link).

[4] R. Nath, P.K. Sehgal, and A.K. Sethi, “Effect of routing misbehavior in mobile ad hoc network,” in *Proc. of Second IEEE Advance Computing Conference*, pp. 218-222, 19-20 February 2010. Article (CrossRef Link).

[5] S.N. Pari and D. Sridharan, “Mitigating routing misbehavior in self organizing mobile ad hoc network using K-neighbourhood local reputation system,” in *Proc. of IEEE International Conference on Recent Trends in Information Technology*, pp. 313-317, 3-5 June, 2011. Article (CrossRef Link).

[6] A. Visconti, and H. Tahayori, “Detecting misbehaving nodes in MANET with an artificial immune system based on type-2 fuzzy sets,” in *Proc. of International Conference for Internet Technology and Secured Transactions*, pp. 1-2, 9-12 November, 2009. Article (CrossRef Link) .

[7] S. Usha and S. Radha, “Co-operative approach to detect misbehaving nodes in MANET using multi-hop acknowledgement scheme,” in *Proc. of International Conference on Advances in Computing, Control & Telecommunication Technologies*, pp. 576-578, 28-29 December, 2009. Article (CrossRef Link).

[8] R. Talreja and V. Jethani, “A vote based system to detect misbehaving nodes in MANETs,” in *Proc. of IEEE International Advance Computing Conference*, pp. 391-394, 21-22 February, 2014. Article (CrossRef Link).

[9] V.N. Patil and S. A. Thorat, “Cross layer approach to detect malicious node in MANET,” in *Proc. of Fourth IEEE International Conference on Computing, Communications and Networking Technologies*, pp. 1-6, 4-6 July, 2013. Article (CrossRef Link).

[10] L. Buttyan and J.P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," *First ACM International symposium on Mobile Ad hoc Networking and Computing*, pp. 87-96, 2000. Article (CrossRef Link).

[11] Zhong, S., Chen, J. and Yang, Y.R. "Sprite: a simple, cheat-proof credit-based system for mobile ad-hoc networks," in *Proc. of Twenty-Second Annual Joint International Conference of the IEEE Computer and Communications, IEEE INFOCOM '03,* vol. 3, pp. 1987-1997, 30 March- 3 April, 2003. Article (CrossRef Link).

[12] L. Buttyan and J.P. Hubaux, "Stimulating cooperation in self-organizing ad hoc networks," *Mobility Networks and Application*, vol. 8, no. 5, pp. 579-592, October, 2003. Article (CrossRef Link).

[13] Y. Xue and K. Nahrstedt, "Providing fault-tolerant ad-hoc routing service in adversarial environments," *Wireless Personal Communications, Special Issue on Security for Next Generation Communications,* vol. 29, no. 3-4, pp. 367-388, June, 2004. Article (CrossRef Link).

[14] M. Conti, E. Gregori and G. Maselli, "Towards reliable forwarding for ad hoc networks," in *Proc. of Eight International Conference on Personal Wireless Communications*, pp. 790-804, 23-25 September, 2003. Article (CrossRef Link).

[15] K. Liu, J. Deng, P. K. Varshney and K. Balakrishnan , "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Transactions on Mobile Computing*, vol.6, no. 5, pp. 536-550, May, 2007. Article (CrossRef Link).

[16] S. Buchegger and J.Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks)," MobiHOC 2002: *IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing*, pp. 226-236, 2002. Article (CrossRef Link).

[17] P. Michiardi. and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks," in *Proc. of Sixth International Federation for Information Processing Conference on Security Communications and Multimedia Security*, pp. 107-121, 26-27 September, 2002. Article (CrossRef Link).

[18] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," *Technical Report*, Stanford University, arXiv:cs.NI/0307012 v2, 2003.

[19] J. Hu and M. Burmester, "LARS: A locally aware reputation system for mobile ad hoc networks," in *Proc. of the Forty Fourth annual Southeast Regional Conference*, pp. 119-123, 10-12 March, 2006. Article (CrossRef Link).

[20] J.N. Al-Karaki and A.E. Kamal, "Stimulating node cooperation in mobile ad hoc networks," *Wireless Personal Communication*. vol. 44, no. 6, pp. 219-239, 2008. Article (CrossRef Link).

[21] K. Gopalakrishnan and R. Uthariaraj, "Local monitoring based reputation system with alert to mitigate the misbehaving nodes in mobile ad hoc networks," in *Proc. of International Conference of Information and Communication Technologies, CCIS*, vol. 101, pp. 344-349, 7-9 September 2010. Article (CrossRef Link).

[22] C. Shi-Hong, L. Chi-Chun and H. Chun-Chieh, "Mitigating routing misbehavior in Dynamic Source Routing protocol using trust-based reputation mechanism for wireless ad-hoc networks," in *Proc. of IEEE Consumer Communications and Networking Conference*, pp. 442-446, 9-12 January, 2011. Article (CrossRef Link).

[23] T. Issariyakul and E. Hossain, "Introduction to network simulator NS2," 2nd Ed. USA, Springer, 2009. Article (CrossRef Link).

Anjali Anand is pursuing Ph.D. from Department of Computer Engineering, Punjabi University, Patiala. She has done M.Tech. from Punjabi University, Patiala. She has contributed 5 articles in various research journals. Her areas of interest are Computer Networks, Mobile Ad hoc Networks. Anjali Anand can be contacted at: anjalianand_87@yahoo.in

Dr. Rinkle Rani is working as Assistant Professor in Computer Science and Engineering Department, Thapar University, Patiala since 2000. She has done her Post graduation from BITS, Pilani and Ph.D. from Punjabi University, Patiala in the area of Computer Networks. She has more than 18 years of teaching experience. She has supervised 34 M.Tech. Dissertations and contributed 50 articles in Conferences and 41 papers in Research Journals. Her areas of interest are Computer Networks and Big data mining and Processing. She is member of professional bodies: ACM, IEEE, ISTE and CSI. She may be contacted at: raggarwal@ thapar.edu

Dr. Himanshu Aggarwal Ph.D., is currently serving as Professor in Department of Computer Engineering at Punjabi University, Patiala. He has more than 22 years of teaching experience and served academic institutions such as Thapar Institute of Engineering & Technology, Patiala, Guru Nanak Dev Engineering College, Ludhiana and Technical Teacher's Training Institute, Chandigarh. He is an active researcher who has supervised more than 30 M.Tech. Dissertations and contributed 80 articles in various Research Journals. He is guiding PhD to 8 scholars and 5 have completed their PhD. He is on the Editorial Board of 9 Journals and Review Boards of 5 Journals of repute. His areas of interest are Software Engineering, Computer Networks, Information Systems, ERP and Parallel Computing. imanshu Aggarwal can be contacted at: himanshu.pup@gmail.com