CHAPTER 4

REPUTATION AND COLLABORATION THROUGH
ROUTE-REQUEST

4.1 Introduction

The performance of mobile ad hoc networks (MANETs) depends upon co-operation among nodes. But limitation of available resources creates an urge among the nodes to misbehave and exhibit non-cooperation for preserving their resources. In MANETs, presence of misbehaving nodes (such as selfish, malicious and erroneous 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 such routing misbehaviour and alleviates its effect.

4.2 Misbehaviour Probability Model

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

A route is considered as ‘misbehaving’ if out of all the nodes in the route at least one node is found to be misbehaving. Routes in the network between the source and the destination have been studied. A route consists of average number of $\delta$ nodes from the source to the destination. Therefore, the number of intermediate nodes in a route is $\delta - 2$ (except the source and the destination). A transmission is considered successful if the route from source to destination does not contain any misbehaving nodes.

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:

$$\text{Prob}_r = 1 - (1 - \text{Prob}_m)^{\delta - 2} \tag{4.1}$$

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

$$\text{Prob}_s = 1 - \text{Prob}_r \tag{4.2}$$

or

$$\text{Prob}_s = (1 - \text{Prob}_m)^{\delta - 2} \tag{4.3}$$

Table 4.1 shows the probability of successful transmission by varying the probability of misbehaving nodes in the network by keeping the average number of nodes in a route $\delta$ constant. From Table 4.1, it can be concluded that the probability of a misbehaving route, $\text{Prob}_r$, increases with an increase in the probability of misbehaving nodes $\text{Prob}_m$. It can be seen that almost half of the routes may become futile when the probability of misbehaving nodes $\text{Prob}_m$ is 0.2. The probability of delivering a packet successfully to the destination $\text{Prob}_s$ also decreases with an increase in the probability of misbehaving nodes. It has been noticed that the probability of successfully delivering a packet $\text{Prob}_s$ is around 0.51 when the probability of misbehaving node $\text{Prob}_m$ is 0.2.

**Table 4.1: Probability of Successful Transmission for varying $\text{Prob}_m$ in the network**

<table>
<thead>
<tr>
<th>$\text{Prob}_m$</th>
<th>$\text{Prob}_r$</th>
<th>$\text{Prob}_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.27</td>
<td>0.73</td>
</tr>
<tr>
<td>0.2</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>0.3</td>
<td>0.34</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Table 4.2 displays the probability of successful transmission by increasing $\delta$ i.e. the average number of nodes in a route, and keeping $\text{Prob}_m$ as constant. As the size of network increases, value of $\delta$ also increases. The probability of successful transmission $\text{Prob}_s$ is inversely proportional to $\delta$. Therefore, the probability of successful transmission $\text{Prob}_s$ decreases with an increase in the value of $\delta$. 
Table 4.2: Probability of Successful Transmission for varying number of nodes in a route

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$Prob_r$</th>
<th>$Prob_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.36</td>
<td>0.64</td>
</tr>
<tr>
<td>6</td>
<td>0.59</td>
<td>0.41</td>
</tr>
<tr>
<td>8</td>
<td>0.73</td>
<td>0.27</td>
</tr>
</tbody>
</table>

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.

### 4.3 Detecting and Handling 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 (Bansal and Baker, 2003) and LARS (Hu and Burmester, 2006), nor does it aggressively spread second-hand information about the behaviour of nodes in the network, like CONFIDANT (Buchegger and Le Boudec, 2002). It lays more emphasis on direct observation but does not completely ignore the opinion of other nodes in the network. Unlike existing schemes such as Weeks and Altun (2006), Tan (2011), Gopalakrishnan and Rhymend Uthariaraj (2011), Wang et al. (2012), Subramaniyan et al. (2014) it creates a concoction of both direct observation and second-hand reputation collected through Route-Request 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 work.

- 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.

### 4.3.1 Lists used in the scheme

The proposed scheme uses 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 address or ID of the accusing node and the $m_{id}$ field contains the address 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 address 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.

### 4.3.2 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 does not forward the packet within 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 packet 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 each of its neighbour nodes. The rating of a node is initialized 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.

$$R = \begin{cases} 
R + R_{inc}, & \text{if positive event} \\
R - R_{dec}, & \text{if negative event}
\end{cases}$$  \hspace{1cm} (4.4)

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 being forwarded by the neighbour node, it compares 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.

### 4.3.3 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 misbehaving threshold $Th_m$, i.e., $R \leq Th_m$, it is considered
as misbehaving and is added to the List $L_M$.

### 4.3.3.1 Using Ancillary List

A variable-length field is added to the 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. Whenever an intermediate node receives a DSR Route-Request packet, it appends its own List $L_A$ to the packet and re-broadcasts it.

Figure 4.1 illustrates how the Ancillary List $L_A$ is appended to the DSR Route-Request packet.

![Figure 4.1: Appending Ancillary List to DSR Route-Request packet](image)

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. Figure 4.2 shows the 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 rebroadcasts 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., $frequency \geq Th_f$, 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. Figure 4.3 illustrates the actions performed by an intermediate node on receiving a Route-Request packet. $R_{Dubious}$ 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

![Figure 4.3: Actions performed by the Intermediate node](image)
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.

**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 greater than or equal to Threshold Frequency $Th_f$ i.e., $frequency \geq Th_f$, then its rating $R$ is set to $R_{Dubious}$. A route reply is generated and sent back to the sending node.

Figure 4.4 illustrates the actions performed by the receiver node on receiving a Route-Request.
**Route-Request Drop**  When an intermediate node receives a Route-Request packet, it checks the Ancillary List $L_A$ in the RREQ packet. 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 Route-Request packet and re-broadcasts it.

Figure 4.5 shows the communication between a source and destination and Node $N_i$ depicts an intermediate node. Intermediate node $N_i$, on receiving the Route-Request packet, checks the intersection between Route-Request route and the List $L_A$. Depending upon the intersection of List $L_A$ received in the DSR Route-Request packet and the DSR RREQ route, node either forwards or drops the Route-Request packet. If the intersection is non-void then the Route-Request is dropped, else it is forwarded to other nodes.

![Figure 4.5: Dropping of Route-Request by an Intermediate node](image)

Algorithm 4.1 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 $T_{hf}$.

**Algorithm 4.1 Using Ancillary List**

1: While (End of List $L_A[h_id][m_id]$)
2: if ($(h_id \neq$ Examining Node’s address) & & $(m_id \neq$ Misbehaving Node’s address)) then
3:    $h_id \leftarrow$ Examining Node’s address
4:    $m_id \leftarrow$ Misbehaving Node’s address
5:    $Frequency \leftarrow 1$
6: else if ($(h_id =$ Examining Node’s address) & & $(m_id =$ Misbehaving Node’s address)) then
7:    Return
8: else
9:    $Frequency \leftarrow Frequency + 1$
10: if ($Frequency \geq T_{hf}$) then
11:    $R \leftarrow R_{Dubious}$
12: end if
13: end if

**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 under the guise of forwarding 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.

**4.3.4 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. On the other hand, a node loses credit with a node upon asking it to forward a packet. When a node receives a request for forwarding by a neighbour, it 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 $T_{Cr}$.

\[ C_{Cr} = C_{Cr} + I, \text{ when } T_{Cr} \text{ expires } \]  

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

4.3.5 Handling Misbehaviour due to failure

A node may wish to forward packets of other nodes’ but could 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, transient link failures, or restarting the network interface. In order to handle such kind of problems, a mechanism known as Reprieval Mechanism has been used in the proposed scheme.

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 the List $L_M$ 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.

4.4 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.
### 4.4.1 Simulation Environment and Performance Metrics

The network simulator NS-2 (version 2.29) (Issariyakul and Hossain, 2008) 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. Twenty simulations have been performed to achieve 95% confidence level and the average value for each data point has been plotted. The source and the destination nodes have been randomly selected among all the nodes in the network. Table 4.3 lists the constant and fixed parameters used in the simulation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{Neutral}$</td>
<td>0</td>
</tr>
<tr>
<td>$R_{inc}$</td>
<td>1</td>
</tr>
<tr>
<td>$R_{dec}$</td>
<td>2</td>
</tr>
<tr>
<td>$T_{PW}$</td>
<td>1 sec</td>
</tr>
<tr>
<td>$T_{hm}$</td>
<td>-40</td>
</tr>
<tr>
<td>$T_{hf}$</td>
<td>3</td>
</tr>
<tr>
<td>$R_{Dubious}$</td>
<td>-30</td>
</tr>
<tr>
<td>$T_{Rep}$</td>
<td>30 sec</td>
</tr>
<tr>
<td>$I$</td>
<td>1</td>
</tr>
<tr>
<td>$T_{Cr}$</td>
<td>10 sec</td>
</tr>
<tr>
<td>Area</td>
<td>1000 X 1000 m$^2$</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>60 Nodes</td>
</tr>
<tr>
<td>Radio Range</td>
<td>250 m</td>
</tr>
<tr>
<td>Speed</td>
<td>0 to 30 m/s (uniformly distributed)</td>
</tr>
<tr>
<td>Sending Capacity</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Packet Size</td>
<td>512 B</td>
</tr>
<tr>
<td>Traffic</td>
<td>CBR</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>900 sec</td>
</tr>
</tbody>
</table>

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 number of control packets generated to the number of data packets sent. Routing Overhead ratio signifies the number of control packets generated for each data packet sent; hence its value should not be considerably greater than that of normal DSR.
• Throughput is defined as the number of data packets correctly delivered to the destination in an observed duration of time.

### 4.4.2 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:

Figure 4.6 illustrates the Packet Delivery Ratio of the proposed scheme and above mentioned algorithms by varying the probability of misbehaving nodes $\text{Prob}_m$. The proposed scheme outperforms other reputation based schemes in terms of packet delivery ratio when $\text{Prob}_m$ is less than equal to 0.5. But when $\text{Prob}_m$ 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.

![Figure 4.6: Packet Delivery Ratio of the proposed scheme and other reputation based schemes](image)

Figure 4.7 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 $\text{Prob}_m$ in-
creases. This is due to the rejection of traffic originating from misbehaving nodes by other nodes in the network. The next best performing algorithm is LMRSA whereas; Normal-DSR protocol shows the worst performance due to absence of any kind of security mechanism.

![Throughput of Misbehaving Nodes](image)

**Figure 4.7: Throughput of misbehaving nodes in the proposed scheme and other reputation schemes**

Figure 4.8 compares an important aspect of Routing Overhead in the proposed scheme with other existing 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, larger number of messages are generated in the network thus; leading to increased routing overhead. The routing overhead of OCEAN is low as it does not spread information about misbehaviour.

Figure 4.9 illustrates the throughput of misbehaving nodes in the proposed scheme for varying misbehaving threshold $T_{hm}$. It can be observed that the proposed scheme has shown a decrease in the throughput of misbehaving nodes as $T_{hm}$ increases. This is due to quick addition
Figure 4.8: Routing Overhead in the proposed scheme and other reputation based schemes

of misbehaving nodes to $L_M$. However, in the proposed scheme, optimum value of $Th_m$ is chosen i.e. -40 to ensure efficient handling of misbehaving nodes and avoid false detection.

Figure 4.9: Throughput of misbehaving nodes in the proposed scheme for varying misbehaving threshold $Th_m$

Figure 4.10 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 either 1 or 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 within 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.

![Figure 4.10: Threshold Frequency of proposed scheme](image)

**4.5 Chapter Summary**

This chapter 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.

The next chapter presents an adaptive security model for handling dynamic behaviour of nodes in mobile ad hoc networks.

**Research Publication out of this Chapter**