# An Adaptive Security Model for Dynamic Node Behaviour in MANETs

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

*Received March 6, 2017; revised September 7, 2017; revised December 23, 2017; accepted January 2, 2018; published June 30, 2018*

## Abstract

Mobile Ad hoc Networks (MANETs) have become a viable platform owing to their potential of providing communication without any pre-existing infrastructure and central administrating authority. Mutual support and co-operation among nodes are prerequisites for performing functions in such networks. The scarcity of resources makes it economical for nodes to conserve their resources and misbehave by avoiding participation in the network. Therefore, a mechanism is required to detect and handle such misbehaving nodes and promote co-operation in the network. Existing techniques for handling misbehaving nodes focus only on their current behaviour without considering the antecedent behaviour of nodes. In real world, a node may dynamically change its behaviour in accordance to its requirements. Hence, an efficient mechanism is required for providing security against such misbehaviour. This paper proposes an Adaptive Security Model which contemplates the present as well as anterior behaviour of nodes for providing security against dynamic node behaviour. The adaptivity of the model is nested in its ability to requite well-behaving nodes and penalize misbehaving ones in conformity with their overall behaviour. Simulation results indicate the efficiency of proposed scheme in securing the network from the menace of dynamic behaviour of nodes.

***Keywords:*** MANET security, dynamic behaviour, reputation based scheme, secure routing, misbehaving nodes, routing misbehaviour.

## 1. Introduction

A mobile ad hoc network (MANET) [1][2][3] is a group of mobile nodes or routers communicating through wireless channel without any fixed infrastructure. These networks provide communications in the absence of central administrating authority without any predefined infrastructure, making them suitable for a variety of applications from critical situations like military and rescue operations, disaster recovery to commercial applications such as environmental monitoring, mobile conferences, etc. Salient features of MANETs include dynamic topology, rapid deployment, flexibility and adaptability.

Nodes, in MANETs, communicate directly with their neighbour nodes within their transmission range through wireless medium. In order to communicate with nodes outside their transmission range, nodes rely on intermediate nodes (each acting as routers) to forward their packets to the destination. Routes between source and destination nodes are established through various routing protocols [4] such as Dynamic Source Routing (DSR) Protocol, Ad hoc On-demand Distance Vector (AODV) protocol etc. DSR [5] and other protocols assume that all the participating nodes will co-operate. The performance of MANETs relies on the consociation of nodes in the network. But there exists a trade-off between co-operation among nodes and consumption of resources. Performing network functions on behalf of other nodes like routing and forwarding may consume a lot of resources such as bandwidth, CPU time and memory, battery power. Therefore, there can be a strong urge for nodes to conserve their resources, while using other nodes' services to relay their own traffic. Such nodes are termed as misbehaving nodes. A node may exhibit different forms of misbehaviour with an objective of conserving its resources. Such kind of misbehaviour or security threats can paralyse or may completely disrupt the network operations [6]. Networks similar to MANETs, like wireless sensor networks can also be affected by such security threats [7][8][9].

Several researchers[3][6][10][11][12][13] have worked in the direction of handling routing misbehaviour and encouraging the misbehaving nodes to co-operate and participate in the network. The existing schemes examine only the current behaviour of nodes without considering their past behaviour. They do not take into account the dynamically changing behaviour of nodes. In real world scenarios, the behaviour of a node may not remain consistent. A misbehaving node may not always misbehave and a well behaving node may not always exhibit good behaviour. Hence, it is essential to devise a mechanism which adaptively handles the varying behaviour of nodes. This paper proposes an Adaptive Requiting and Punitive Mechanism which introduces the following novel features in comparison to pioneer researches [14][15][16] for dealing with node misbehaviour:

1. The proposed ARPM scheme contemplates the present as well as anterior forwarding behaviour of nodes for securing MANETs against misbehaviour.
2. It provides an adaptive security model for efficient handling of dynamic node behaviour.
3. It provides a requiting mechanism which gives incentives to nodes based on their past forwarding behaviour.
4. It provides a punitive mechanism for penalizing misbehaving nodes based on prior communications and their frequency of misbehaviour.
5. It gives differential treatment to non-participating and nodes lying on the periphery of the network which results in increased throughput.

The next section describes the other related works in the field followed by Problem Statement discussion in Section 3. The details about the components of the proposed scheme are

presented in Section 4. Section 5 discusses the implementation details and simulation environment used in the proposed work. The last section includes conclusion along with the future scope of the proposed work.

## 2. Related Works

Several approaches have been given in the literature for handling misbehaving nodes in mobile ad hoc networks. These approaches can be broadly divided into two categories: Credit-based or Incentive-based schemes [17][18][19] and Reputation-based schemes [14][15][20][21][22][23][24][25][26][27][28][29]. In credit-based schemes, nodes are given incentives to provide services to others. On the other hand, reputation-based schemes are based on observation by the nodes. These schemes mainly fall into two categories: local and global. In local reputation-based schemes, a node makes an assessment about the behaviour of another node by directly communicating with it. On the other hand, in global reputation-based schemes, a node gathers information about the behaviour of a node from other nodes in the network.

CONFIDANT [20] uses direct observation and alarm signals from other nodes for detection of misbehaving nodes. The weightage given to alarm signals depend upon the trustworthiness of the sending node. CONFIDANT uses Bayesian Estimation Model for calculating trust relationship and reputation of nodes which increases the complexity of the system and it is vulnerable to rumour spreading.

CORE [21] is a reputation mechanism to promote node co-operation in MANETs. It uses a combination of three types of reputation values (direct, indirect and functional) to make decision for either including or excluding the node from the network. It does not provide any second-chance mechanism to the nodes that are excluded from the network which can be quite harsh on the nodes which were unable to participate due to some system failure.

OCEAN [14] depends solely upon the direct observation of their neighbours for maintaining reputation information. It adds the misbehaving nodes into a `Faulty List' and all communications with them are stopped. It uses a `Second Chance Mechanism' which provides `faulty nodes' an opportunity to improve their behaviour. One of the limitations of OCEAN is that it lacks any adaptive mechanism for handling dynamic behaviour of nodes.

LARS [15] is `Locally Aware Reputation System' for mitigating selfish misbehaviour of nodes. It uses direct observation approach for calculating the reputation of nodes in the network. A node notifies other nodes about a node's misbehaviour by spreading a WARNING message in the network which increases the routing overhead.

Al-Karaki and Kamal [22] have proposed a fully distributed scheme for stimulating node co-operation which uses a combination of reputation-based with virtual currency based techniques. This mechanism does not provide any opportunity to misbehaving nodes to improve as their misbehaviour may be due to some kind of failure in the system. Moreover, the information about the misbehaving node is spread in the network which increases the network traffic.

LMRSA [16] is a local monitoring reputation sytem which uses only direct observation for maintaining reputation of nodes in the network. This scheme isolates the misbehaving nodes from the network upon their detection. A node which detects a misbehaving node in the route sends an ALERT message to the source node for informing about the misbehaviour. Upon receiving an ALERT message, the source node removes the route consisting of the misbehaving node from its route cache.

Chen et al. [23] have proposed FITS which is a reputation system that is based on Subgame Perfect Nash Equilibrium. It uses a Threat To Interfere (TTI) technique that allows a node to threaten other nodes if they refuse to co-operate in the last stage of finite game. However, the limitation of this scheme it has assumed a fixed topology network which is impractical in real-world scenarios where the topologies change dynamically.

Noorian et al. [24] have proposed a reputation mechanism based on `Prisoner Dilemma' game theory. Nodes determine the `expected payoff' of different forwarding behaviours towards its neighbours and choose the most suitable one with highest payoff. The limitation of this approach is that it does not consider the varying misbehaviour of nodes.

Paramasivan et al. [25] have proposed a Perfect Bayesian Estimation (PBE) technique for making a decision about the type of node (regular or malicious) and reporting it to other nodes. The drawbacks of this technique are: increased complexity due to use of Bayesian model and increased traffic overhead due to information sharing. Moreover, it deals with only packet-dropping misbehaviour which may be insufficient for comprehensive security against routing misbehaviour.

The proposed scheme provides an Adaptive Requiting and Punitive Mechanism (ARPM) for secure routing in mobile ad hoc networks. The mechanism is adaptive in the sense that it takes into consideration the prior communication between the nodes for dealing with misbehaviour in the network. **Table 1** highlights the distinguishing features of the proposed scheme and other existing direct observation based reputation schemes. In these schemes nodes monitor the behaviour of other nodes through direct communication and use only first-hand information for maintaining reputation of nodes. Unlike OCEAN [14], LARS [15] and LMRSA [16], the proposed ARPM scheme gives differential treatment to non-participating nodes and peripheral nodes. ARPM analyses the anterior forwarding behaviour of nodes to requite the good behaviour of nodes and to penalize the misbehaving ones.

**Table 1.** Comparison of the proposed ARPM with other direct observation based reputation schemes

| FEATURES | OCEAN | LARS | LMRSA | PROPOSED ARPM |
|---|---|---|---|---|
| Direct Observation | Yes | Yes | Yes | Yes |
| Differentiation between non-participating and peripheral nodes | No | No | No | Yes |
| Adaptive in nature | No | No | No | Yes |
| Handles On/Off Misbehaviour | No | No | No | Yes |
| Consideration given to anterior forwarding behaviour of nodes | No | No | No | Yes |
| Requites good behaviour of nodes | No | No | No | Yes |
| Penalizes bad behaviour of nodes | Yes | Yes | Yes | Yes |

## 3. Problem Statement

This section discusses the problem of routing misbehaviour in mobile ad hoc networks and the effect of misbehaving nodes on the performance of the network.

### 3.1 Routing Misbehaviour Problem

Nodes in MANETs depend upon co-operation among nodes for performing network functions such as routing packets from source to destination. However, due to limitation of resources, nodes may decide not to co-operate in order to save their resources. Such behaviour is known as routing misbehaviour. In real world scenarios, the behaviour of a node can change dynamically. A node exhibiting dynamic behaviour may change its behaviour several times depending upon its requirement. It may show normal behaviour at certain times or misbehave at other times. A dynamically behaving node, when misbehaving, can exhibit different kinds of routing misbehaviour. The proposed scheme is capable of detecting and handling the following types of misbehaviour:

- *Packet Drop*: A node responds positively to route requests of other nodes but then drops packets rather than forwarding them and misleads other nodes to successfully send their traffic through it.
- *Non-participation*: A node may forward its own traffic through other nodes but restraint other nodes in the network to send their traffic through it. It aims at saving its resources such as battery power by denying forwarding requests of other nodes. It limits its participation to just sending and receiving its own packets.
- *On-Off Misbehaviour*: A node may change its behaviour according to the circumstances at any time. It may forward a fraction of data packets for some time and then may start dropping packets. It may resume its normal behaviour depending upon its own requirements and may misbehave several times.
- *Unintentional Packet Drop*: A node may wish to forward packets for other nodes but is unable to do so due to failure. It has no intention of dropping packets of other nodes but fails to forward 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 considered to be dropped.
- *Tampering of data packets*: A node may participate in route discovery and route maintenance process but once the route is established and packets are being transmitted, it may tamper with the data during the forwarding process.

### 3.2 Effect of Misbehaving Nodes on Network Performance

This sub-section demonstrates how the presence of misbehaving nodes deteriorates the network performance. Routes in the network between the source and the destination have been studied. A route consists of an average number of '*k'* hops in a transmission. The probability that a node may misbehave irrespective of the type of misbehaviour, is given by $P_M$. The number of misbehaving nodes in the network is represented by $N_M$ and the total number of nodes in the network is given by $N_{Total}$. Therefore, the probability that a packet will be successfully delivered to the destination is given by:

$$P_{Success} = \frac{\begin{bmatrix} N_{Total} - (round(k) - 1) \\ N_M \end{bmatrix}}{\begin{bmatrix} N_{Total} \\ N_M \end{bmatrix}} \tag{1}$$

Equation (1) represents the probability of ways in which nodes in the network can be misbehaving provided that none of the nodes in the k-hop route is misbehaving. Hence, the transmission through the '*k*' hop route will be successful as no node in the route is misbehaving. For estimating $P_{Success}$, the average number of hops in a route '*k*' must be known.

The average number of hops in a route *k* are estimated using the methodology given in [30]. Let '*p*' be the average progress of each hop in the network and the average distance between source and destination be '*d*'. Then '*k*' can be estimated as:

$$k = \frac{d}{p} \tag{2}$$

The value of one-hop progress '*p*' can be estimated as the average of distance between the source and each node lying within its range. Let $\gamma$ be the average number of nodes within the transmission range '*R*' of a node in the network. Then, $\gamma$ can be expressed as:

$$\gamma = \frac{N_{Total}}{X * Y} \cdot \pi R^2 \tag{3}$$

where $X * Y$ represents the network area and the value of $\pi R^2$ represents the area within the transmission range of a node. $\gamma$ is assumed to be an integer for simplification. Let *F(r)* represents the probability of all $\gamma$ nodes lying within the area '$\pi r^2$' of the source node.

$$F(r) = \left[\frac{\pi r^2}{\pi R^2}\right]^{\gamma} = \frac{r^{2\gamma}}{R^{2\gamma}} \tag{4}$$

The Probability Density Function (PDF) of hop-progress *'r'* from source node can be defined as:

$$f(r) = \frac{\partial}{\partial r}(F(r)) = \frac{2E \cdot r^{2\gamma-1}}{R^{2\gamma}} \tag{5}$$

The average progress can be estimated as the probable value of '*r*' with respect to probability density function *f(r)*:

$$p = \int_0^R r f(r) dr = \frac{2\gamma \cdot R}{2\gamma + 1} \tag{6}$$

Let the distance between the source and destination node be estimated as:

$$d \approx \frac{(0 + \sqrt{X^2 + Y^2})}{2}$$

Therefore, the average number of hops '*k*' in a route can be determined as:

$$\frac{d}{p} = \frac{\sqrt{X^2 + Y^2}}{2p} = \frac{(2\gamma + 1)\sqrt{X^2 + Y^2}}{4\gamma R} \tag{7}$$

Combining equation (1), (2) and (7), the Probability of Successful Transmission is given by:

$$P_{Success} = \frac{\begin{bmatrix} N_{Total} - (\frac{(2\gamma+1)\sqrt{X^2+Y^2}}{4\gamma R} - 1) \\ N_M \end{bmatrix}}{\begin{bmatrix} N_{Total} \\ N_M \end{bmatrix}} \tag{8}$$

where '$\gamma$' is given by equation (3). **Table 2** illustrates the Probability of Successful Transmission, $P_{Success}$ for varying number of misbehaving nodes in the network. From **Table 2**,

it can be concluded that the Probability of successful transmission $P_{Success}$ decreases with an increase in $P_M$. It also decreases with an increase in the size of the network as routes become longer.

**Table 2.** Probability of successful transmission, $P_{Success}$ for varying number of misbehaving nodes in the network, $N_M$

| $P_M$ = 0.1 | | | |
|---|---|---|---|
| **Network Area** | 3R*3R | 5R*5R | 10R*10R |
| $N_{Total}$ | 50 | 100 | 500 |
| $N_M$ | 5 | 10 | 50 |
| $P_{Success}$ | 0.90 | 0.80 | 0.58 |
| **$P_M$ = 0.2** | | | |
| **Network Area** | 3R*3R | 5R*5R | 10R*10R |
| $N_{Total}$ | 50 | 100 | 500 |
| $N_M$ | 10 | 20 | 100 |
| $P_{Success}$ | 0.80 | 0.65 | 0.40 |
| **$P_M$ = 0.3** | | | |
| **Network Area** | 3R*3R | 5R*5R | 10R*10R |
| $N_{Total}$ | 50 | 100 | 500 |
| $N_M$ | 15 | 30 | 150 |
| $P_{Success}$ | 0.70 | 0.48 | 0.24 |

**Table 2** clearly demonstrates the detrimental effects of misbehaving nodes in MANETs. For example, in a network of size $5R*5R$ where $P_M$ is 0.3, the probability of successful transmission is less than 50 percent. With such a low probability of successful transmission, the performance of the network may significantly degrade. From the above discussion it can be concluded that a high probability of misbehaving nodes in the network can have a profound impact on its performance.

## 4. The Proposed ARPM Scheme

The proposed scheme provides an Adaptive Requiting and Punitive Mechanism (ARPM) for handling dynamic behaviour of nodes. It has been devised to work with Dynamic Source Routing (DSR) Protocol. Its details can be found in [5]. Dynamic behaviour of nodes can easily deceive the existing reputation system, such as OCEAN, by maintaining the ratio between time of misbehaviour and good behaviour according to the defined increment and decrement in reputation values as explained by [31]. Whereas, the proposed scheme contemplates the anterior forwarding behaviour of nodes and the frequency of misbehaviour which makes it difficult for the misbehaving nodes to deceive the reputation system.

The Requiting and Punitive Mechanism handle different types of routing misbehaviour. The Requiting Mechanism handles non-participation misbehaviour whereas; the Punitive Mechanism handles packet-drop, tampering of data packets and On/Off misbehaviour. The mechanism is adaptive in the sense that it takes into consideration the prior communication between the nodes for dealing with dynamic behaviour of nodes. It not only requites well-behaving nodes but also penalizes misbehaving ones. The Requiting Mechanism encourages nodes to participate in the network functions in the form of *credit-chips* which are incremented depending upon the antecedent communication with that node. On the other

hand, the Punitive Mechanism punishes the misbehaving nodes by ostracizing them from the network for a certain period of time. These nodes are included back into the network after timeout. The time period is adaptive in nature and depends upon the frequency of its misbehaviour and the anterior forwarding behaviour of that node. 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.
- Misbehaviour is limited to individual nodes.
- Source and destination are randomly selected for various transmissions.

### 4.1 Components of the Proposed ARPM Scheme

The proposed scheme is deployed at every node in the network. It consists of four components: Monitor, Banker, Adaptive Requiting and Punitive Mechanism (ARPM) and Route Request Ratification. **Fig. 1** illustrates the workflow of the Adaptive Security Model along with its components.

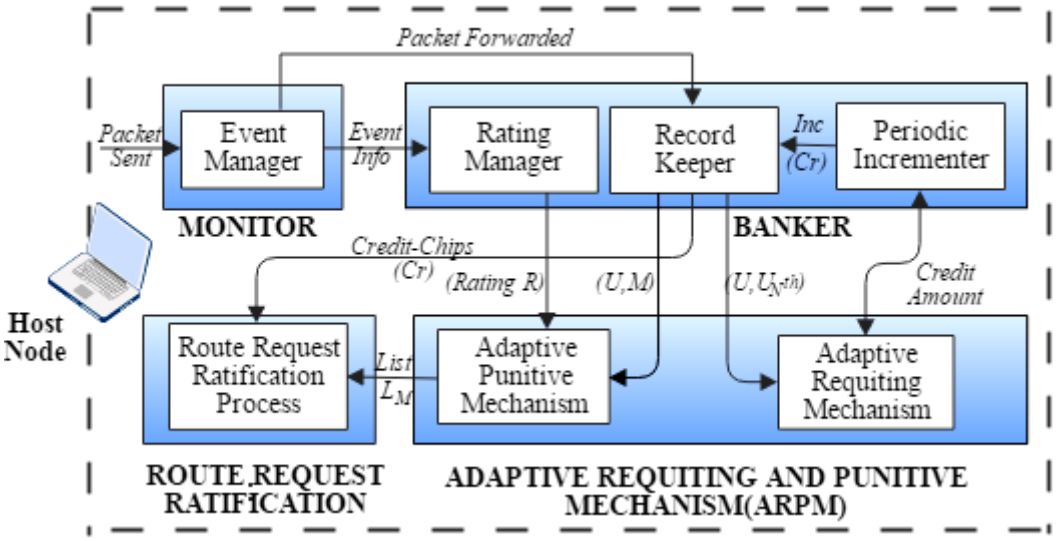


**Fig. 1.** Workflow of the Adaptive Security Model

When the communication begins, Monitor component monitors the neighbour node in promiscuous mode and sends a signal to the Banker when the packet is forwarded or dropped. The Rating Manager increments or decrements the Rating of the neighbour node depending upon the positive or negative event. The Record-keeper module in Banker maintains a record of the number of packets forwarded by each neighbour node of the host node and the packets forwarded by the host node of its neighbour nodes. The Adaptive Requiting and Punitive Mechanism uses these records for calculating the *Credit Amount* and *T(n)* for each neighbour node. The Periodic Incrementer module is responsible for incrementing the *Credit-Chips* for each neighbour node at regular intervals. The Route Request Ratification component accepts or denies a forwarding request by a neighbour node depending upon the information from the Record-Keeper and Adaptive Punitive Mechanism.

### 4.1.1 Monitor

The Monitor component is responsible for monitoring the activities of neighbour nodes. When a packet is forwarded, the Event Manager sub-component saves the checksum of that packet. Once the packet is sent, it overhears the wireless medium for time $T_M$. If the neighbour node attempts to forward a packet, the Monitor compares the checksum of the forwarded packet with the saved checksum. If the checksum matches, a positive event is registered. On the other hand, in case of mismatch packet is considered as being dropped. If there is no successful attempt by the neighbour to forward the packet within time $T_M$, a negative event is registered against it. This information is then conveyed to the Banker.

It monitors not only the behaviour of the neighbouring nodes but also keeps a check on the tampering or modification of the packets before being forwarded by the neighbour node. Upon hearing a packet being forwarded by the neighbour node, it compares the forwarded packet with the checksum saved in its buffer. If the neighbour node tries to modify the contents of the packet, the checksum will not match, and the packet will be treated as not being forwarded or dropped.

### 4.1.2 Banker

The Banker is responsible for accounting all the events and communication between the host node and its neighbour nodes. It consists of three sub-components: Record-keeper, Rating Manager and Periodic Incrementer.

- Record-keeper

The Record-keeper module maintains a record of the number of packets forwarded by neighbour node on behalf of host node *U*, number of packets successfully transmitted by host node of the neighbour node *M* and the *Credit-Chips* available with the neighbour node at the host node i.e., *Cr* for each of its neighbour node. When the neighbour node forwards a packet of the host node, the Record-keeper module updates the value of *U* and *Cr*. On the other hand, when the host node relays a packet of its neighbour node, the record-keeper updates the value of *M*, *Cr*. These updated values are overwritten on the previously stored values in order to save the cache memory. Record-keeper also maintains a record of the number of packets forwarded by the neighbour node on behalf of the host node in the last elapsed time slot i.e., $U_{Nth}$. The value of $U_{Nth}$ is overwritten whenever the next time slot expires.

- Periodic Incrementer

The Periodic Incrementer module in Banker is responsible for updating the value of *Credit-Chips* periodically. It increments the *Credit-Chips* amount for all the nodes in the network at regular intervals when the Bank Timer $T_B$ expires. The increment amount is calculated by Adaptive Requiting Mechanism. If no communication has occurred with the node in the past, then the increment amount is set to Default Bank Amount for that node.

- Rating Manager

Rating Manager, at each node, maintains rating *R* for each of its neighbour nodes. The rating of a node is initialized to $R_{Neutral}$ and is incremented by an amount $R_{inc}$ in case of a positive event and decremented by amount $R_{dec}$ on receiving a negative event from the Event Manager module in the Monitor Component.

### 4.1.3 Adaptive Requiting and Punitive Mechanism (ARPM)

DSR provides no mechanism for detection and handling routing misbehaviour. The Adaptive Requiting and Punitive Mechanism is used for handling different types of routing misbehaviour. It solely depends upon direct observation of nodes for maintaining reputation and avoids the complexity of second-hand information. The Requiting and Punitive Mechanism handle different types of routing misbehaviour. The Requiting Mechanism

handles non-participating misbehaviour whereas, the Punitive Mechanism handles packet-drop (both intentional and unintentional), tampering of data packets and On/Off misbehaviour. The mechanism is adaptive in the sense that it takes into consideration the prior communication between the nodes for requiting and penalizing the nodes in the network.

- **Adaptive Requiting Mechanism**

The Requiting Mechanism is used for handling non-participation misbehaviour. Nodes in the network are requited for participating in the forwarding process in the form of *Credit-Chips*. In this mechanism, each node maintains a counter known as *Credit-Chips* for each of its neighbour node. A node gains *Credit-Chips* at a node when it forwards a packet on behalf of that node. It looses *Credit-Chips* at a node when it requests that node to forward its packets. On receiving a forwarding request from a neighbour node, the node checks the *Credit-Chips* amount for that neighbour node, If the value of *Credit-Chips* is non-zero only then the packet is forwarded else the request is denied. A node must transmit packets of other nodes if it wishes to relay its own traffic through them. Hence, a non-participating node must concede its selfish behaviour or else it will be ostracized from the network.

However, this technique suffers from the problem of being unfair to the nodes on the periphery of the network. Due to their location, the nodes on the periphery of the network may not get sufficient opportunity to forward packets of other nodes. As a result, their *Credit-Chips* may decrease and nodes in the network may stop forwarding their packets due to lack of *Credit-Chips* available with them.. To address this problem, *Credit-Chips* for each node in the network are incremented after a stipulated interval of time, known as *Bank Timer* ($T_B$). The increment amount for each node is calculated after a fixed interval of time, known as *Requite Timer* ($T_R$). The higher the number of packets forwarded, greater will be the increment amount. Hence, the proposed requiting mechanism thwarts the non-participating nodes from relaying their traffic while allowing peripheral nodes to send their data as the number of packets forwarded by them would be greater than that of non-participating nodes. The Requiting Mechanism is responsible for determining the *Credit Amount* given to each node based on its past forwarding behaviour. The increment amount is calculated after a fixed interval of time, known as *Requite Timer* ($T_R$). When the Requite Timer $T_R$ expires, the average number of packets forwarded ($Avg_N$) in $N$ time slots is calculated using $Avg_{N-1}$ and $U_{Nth}$ as given below:

$$Avg_N = \frac{Avg_{N-1} * (N-1) + U_{Nth}}{N} \tag{9}$$

*Normalization Factor (N.F.)* is calculated using the values of $Avg$, $Avg_{\min}$ and $Avg_{\max}$ as shown below:

$$N.F. = \frac{Avg - Avg_{\min}}{Avg_{\max} - Avg_{\min}} \tag{10}$$

Finally, the *Credit Amount* is determined using *Normalization Factor* and *Initial Credit-Chips* amount for each neighbour node as follows:

$$C.A. = Initial\_Credit\_Chips(1 + N.F.) \tag{11}$$

The *Credit Amount* given to a node is calculated using the number of packets forwarded not only in the previous time slot but over all the time slots. Therefore, the averaging technique ensures that even if a node does not get a chance to forward packets in the previous time slot (as in case of peripheral node), its *Credit Amount* will decelerate in accordance with the average number of packets forwarded by the node. Moreover, it also effectively handles the

dynamic behaviour of nodes by keeping a check on the sporadic non-participation. If a node decides to forward packets for another node only to gain *credit-chips* in order to relay its traffic and stops forwarding them when it no longer requires to transfer its data then, the average number of packets forwarded by the node would be low, which in turn will reduce the *Credit Amount* given to it. Algorithm 1 shows the steps involved in the Adaptive Requiting Mechanism and how it calculates the *Credit Amount* for each node from its previous forwarding behaviour. The complexity of the algorithm is $O(c)$, where $c > 0$.

**Algorithm 1: Adaptive Requiting Mechanism**

**when** ($T_R$ expires)
$Avg_N=(Avg_{N-1}*(N-1)+U_{Nth})/N$
  **if** ($Avg_N < Avg_{min}$ ) **then**
    **set** $Avg_{min} = Avg_N$;
  **end**
  **if** ($Avg_N > Avg_{max}$ ) **then**
    **set** $Avg_{max} = Avg_N$;
  **end**
  **if** ($Avg_{min} == 0$ && $Avg_{max}==0$) || $Avg_{min}==Avg_{max}$) **then**
  **set** N.F. = 0;
**else**
  **set** N.F. = ($Avg_N – Avg_{min}$) /( $Avg_{max}$-$Avg_{min}$);
**end**
**set** Credit_Amount = Initial_Credit_Chips (1 + N.F. );
**end**

- **Adaptive Punitive Mechanism**

The Adaptive Punitive Mechanism handles Packet Drop misbehaviour, Tampering of Data Packets and On/Off misbehaviour. Packet Drop misbehaviour could either be intentional or unintentional. This mechanism keeps a check on the Rating $R$ of the neighbour nodes maintained by the Rating Manager in Banker. Rating of a node is decremented by the host node when it drops packets rather than forwarding them. When the Rating $R$ of a node is less than $Th_m$ i.e., $R <= Th_m$, it is considered as misbehaving and is added to the List $L_M$ . Once a node is added to List $L_M$ , all communications with the accused node are stopped.

If a node tries to tamper with the data before forwarding the packet, the checksum will not match and the Monitor will generate a negative event against that node. In case of continued tampering of packets, the Rating of the node will fall below $Th_m$ and the Punitive Mechanism will add it to List $L_M$ blocking all the communications with the respective misbehaving node.

However, blocking the nodes for an indefinite time period could be unfair to nodes exhibiting Unintentional Packet Drop misbehaviour. Such nodes do not have any intention of dropping packets but are unable to forward them due to some kind of failure in their system such as transient link failure, power failure etc. To account for such problems, a time-out based technique is implemented in which a node is withdrawn from the List $L_M$ after a certain period of time. When a misbehaving node is added to List $L_M$, a timer $T_o$ is started. When the timer expires, the node is taken out of List $L_M$ and all communications with it are resumed. As a result, nodes exhibiting Unintentional Packet Drop misbehaviour will be able to continue their operations in the network once they recover from the failure.

However, this technique may fall prey to dynamic behaviour of nodes as they may change their behaviour according to different circumstances. Nodes may exhibit On/Off misbehaviour

by transmitting only small number of packets in order to deceive the monitoring system. A node may forward a few packets and then drop them as and when required. It may exhibit repeated misbehaviour and may enter and leave the List $L_M$ various times. Such behaviour can severely degrade the performance of the network.

To handle such issues, the proposed scheme uses an Adaptive Punitive Mechanism based upon the anterior forwarding behaviour of nodes. Each node is given an Initial Timer $T_o$ when it is added to List $L_M$. The Initial Timer $T_o$ adapts according to the frequency $n$ and Adjustment Factor. Adjustment Factor *A.F.* is given by:

$$A.F. = 1 - \frac{U}{M + U} \tag{12}$$

where $U$ is the number of packets forwarded by neighbour node on behalf of host node and $M$ is the number of packets successfully transmitted by host node of the neighbour node. The timer $T(n)$ given to a node is extended adaptively, each time it enters the List $L_M$ , depending upon the prior communication with that node. It not only depends upon the frequency $n$ of a node entering into $L_M$ but also on the communication between the host node and the accused node. The value of Timer $T(n)$ is given by:

$$T(n) = T_o(1 + n * A.F.) \tag{13}$$

**Table 3** shows the effect of varying $U$ and $M$ on Timer $T(n)$ by keeping frequency $n$ as constant. For simplicity of discussion, the frequency of node misbehaviour '$n$' is set as 3. When the number of packets forwarded by misbehaving neighbour node of the host node, i.e. $U$, is twice the number of packets forwarded by the host node of the misbehaving neighbour node, i.e. $M$, then the time period after which the misbehaving neighbour node is removed from the list $L_M$ of the host node, given by $T(n)$, becomes twice the Initial Timer $T_o$ . The value of $n$=3 indicates that the node is misbehaving for the third time. Consequently, the timer $T(n)$ is doubled despite the higher value of $U$. In the next case, it can be observed that the value of $U$ is half of the value of $M$. Therefore, the value of timer $T(n)$ gets thrice the Initial Timer $T_o$ for the same frequency $n$=3. In the 3rd case, the value of $U$ is equal to the value of $M$, resultantly, the value of timer $T(n)$ is 2.5 times the Initial Timer $T_o$ .

**Table 3.** Effect of varying $U$ and $M$ on timer T($n$)

| ***n* =3** | | |
|---|---|---|
| ***U*** | ***M*** | ***T(n)*** |
| 2x | x | 2 $T_o$ |
| x | 2x | 3 $T_o$ |
| x | x | 2.5 $T_o$ |

Algorithm 2 shows the steps involved in the Adaptive Punitive Mechanism and how it calculates the Timer Value $T(n)$ for each misbehaving node from its previous forwarding behaviour. The complexity of the algorithm is $O(c)$, where $c > 0$.

**Algorithm2: Adaptive Punitive Mechanism**

```
set M= 0; U= 0; n= 0;  T_o = Default Initial Timer; A.F.= 0; T(n)= 0;
if (R >= Th_m)
        Add node to list L_M
        set n= n+1;
        set M= getHostForwardingCount();
        set U= getNeighbourForwardingCount();
            if((U==0) || ( M==0)) then
   set A.F.=1;
            end
            else
   set A.F.=1- (U/(U +M));
            end
            set T(n)= T_o * (1 + (n * A.F.) );
 end
```

### 4.1.4 Route Request Ratification

This component performs the task of rejecting traffic from misbehaving nodes. The policy of traffic rejection is adopted to prevent misbehaving nodes from sending their own traffic under the guise of forwarding them on some other nodes behalf. Upon receiving a route request from a neighbouring node, a node checks its Misbehaving Nodes' List $L_M$. If the requesting node is present in its List $L_M$ then the route request is dropped else it is forwarded. Similarly, on receiving a route reply from a neighbouring node, the node first checks whether the replying node is in its Misbehaving Nodes' List $L_M$ or not. If it is not present in the List $L_M$, only then the reply is accepted else it is rejected.

This component also performs the task of accepting and rejecting a forwarding request by a node based on the *Credit-Chips* available with that node. If the amount of *Credit-Chips* for a neighbour node requesting a forwarding service lies below a certain threshold then the request is denied else, the request is accepted. Therefore, a node must have sufficient *Credit-Chips* in order to relay its own traffic. A node which misbehaves and saves its resources will eventually lose all the *Credit-Chips* at its neighbour nodes and will soon be refused the network services. In order to regain its network services, the node must concede its selfish behavior or else it will remain isolated from the network.

## 5. Simulation Environment and Implementation

This section presents the simulation environment and comparative analysis of the proposed scheme ARPM and other existing direct observation based reputation schemes such as OCEAN, LARS, LMRSA along with traditional DSR protocol in the presence of misbehaving nodes.

### 5.1 Simulation Environment and Performance Metrics

The network simulator NS2 (version 2.29) [32] has been used for running the simulations. Modifications have been done in the DSR module in NS2 to simulate misbehaving nodes. **Table 4** below highlights the fixed parameters used for simulation.

**Table 4.** Fixed parameters for simulation

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Area | 1500 X 1500 $m^2$ | Initial *Credit-Chips* | 10 |
| Number of Nodes | 100 | $R_{Neutral}$ | 0 |
| Mobility Model | Random Waypoint | $R_{inc}$ | 1 |
| MAC | 802.11 | $R_{dec}$ | 2 |
| Speed | 0 to 25 m/s | $Th_m$ | - 40 |
| Range | 250 m | $T_M$ | 500 ms |
| Sending Capacity | 2 Mbps | $T_R$ | 30 sec |
| Traffic | CBR | $T_B$ | 10 sec |
| Simulation Time | 900 sec | $T_o$ | 30 sec |

The performance of proposed ARPM scheme has been evaluated using the following metrics:

- Packet Delivery Ratio is defined as the ratio of number of data packets received by the destination to the number of data packets sent by the source.
- Throughput is defined as the number of data packets correctly delivered to the destination in an observed duration of time.
- Average End-to-End Delay is defined as the time taken for a packet to be transmitted across a network from source to destination.

## 5.2 Simulation Results

This sub-section gives the simulation results obtained for the proposed scheme and other deployed schemes along wih normal DSR protocol. The simulation procedure is repeated 25 times to achieve 95 percent confidence level. Average values are plotted for each data point.

**Fig. 2** displays the Packet Delivery Ratio of the proposed scheme and other deployed schemes. The packet delivery ratio of proposed scheme is greater than that of other existing schemes in the presence of misbehaving nodes. This is due to the ability of ARPM to contemplate the current as well as anterior forwarding behaviours of nodes for detecting and handling misbehaving nodes. It can be observed that the Packet Delivery Ratio decelerates quickly as the probability of misbehaving nodes $P_M$ in the network increases. Networks with such high probability of misbehaving nodes are impractical and must be discarded.

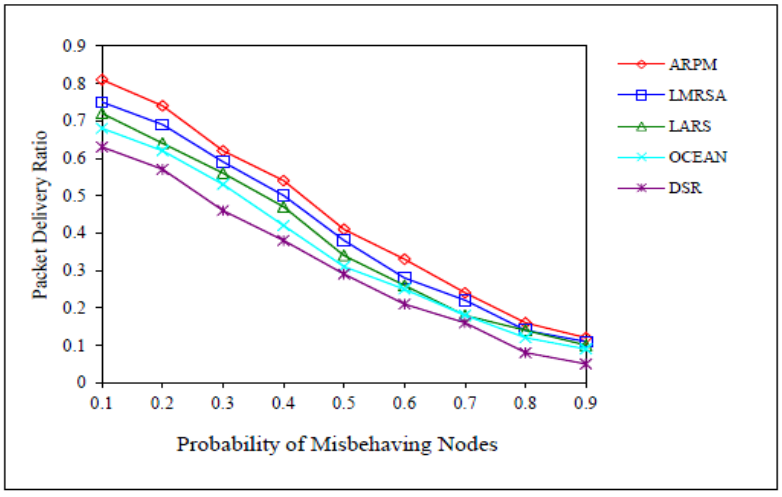


**Fig. 2.** Packet delivery ratio of the proposed scheme and other reputation based schemes

The throughput of misbehaving nodes in the proposed scheme (ARPM) and various other reputation based schemes is illustrated in **Fig. 3**. Throughput of misbehaving nodes, in case of ARPM, is less than all other deployed algorithms. This is due to Adaptive Punitive Mechanism which blocks misbehaving nodes and disallows them to communicate in the network based on their past communication and frequency of their misbehaviour. However, it can be observed that the throughput of all the schemes decreases when the probability of misbehaving nodes $P_M$ is very high as a deadlock situation is created within the network because of large number of misbehaving nodes.

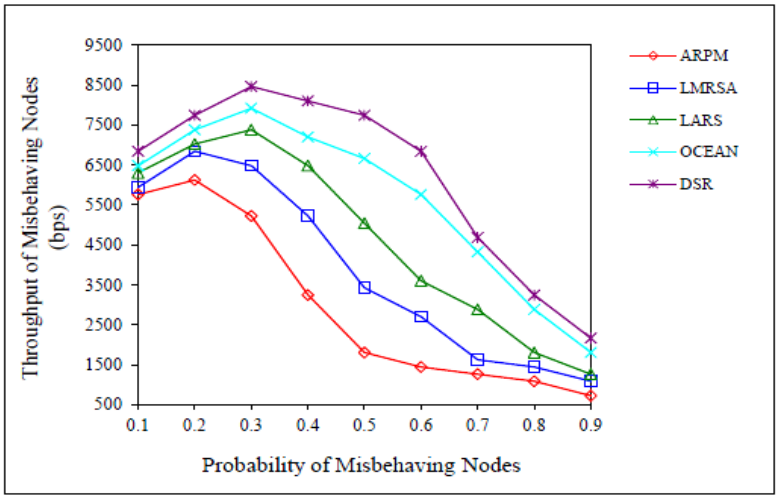


**Fig. 3.** Throughput of misbehaving nodes in the proposed scheme and other reputation schemes

**Fig. 4** represents the routing overhead generated in the proposed scheme (ARPM) and other algorithms. ARPM shows minimum routing overhead as compared to other deployed schemes as it does not generate any kind of explicit messages for spreading misbehaviour information. Whereas LARS and LMRSA use Warning or Alert messages to disseminate information regarding misbehaviour nodes which results in increased control traffic in the network.

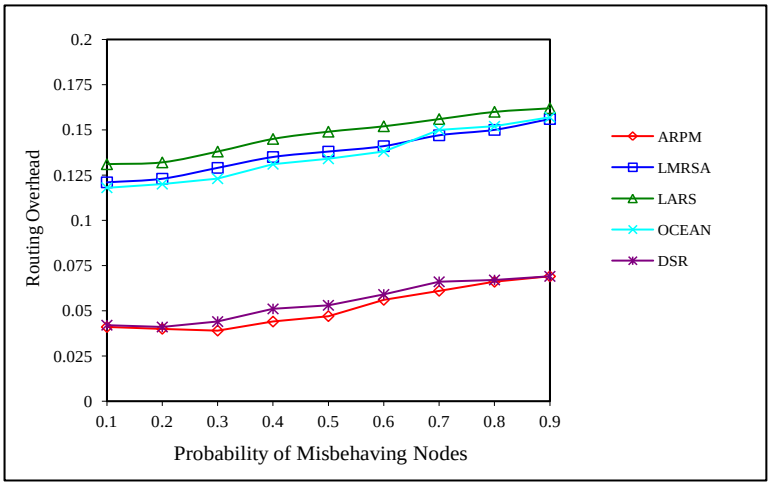


**Fig. 4.** Routing overhead in the proposed scheme and other reputation schemes

**Fig. 5** illustrates the throughput of well-behaving nodes in the proposed ARPM scheme along with other reputation based schemes. The throughput of well-behaving nodes, in ARPM, is high due to its ability to efficiently handle on-off misbehaviour and allowing nodes to by-pass misbehaving ones during route formation. Moreover, it provides differential treatment to non-participating and peripheral nodes which increases throughput in the network. Other deployed schemes provide relatively low throughput as they lack any mechanism for handling on-off misbehaviour.

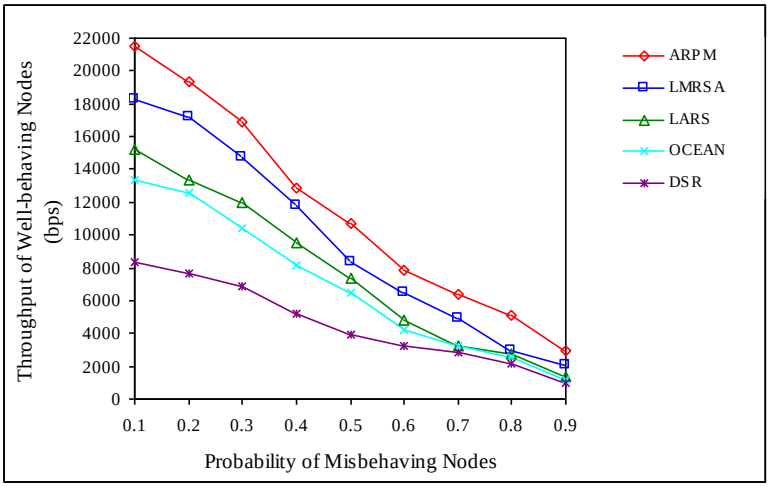


**Fig. 5.** Throughput of well-behaving nodes in the proposed scheme and other reputation schemes

**Fig. 6** shows the Average End-to-End Delay of the proposed scheme and other existing schemes.

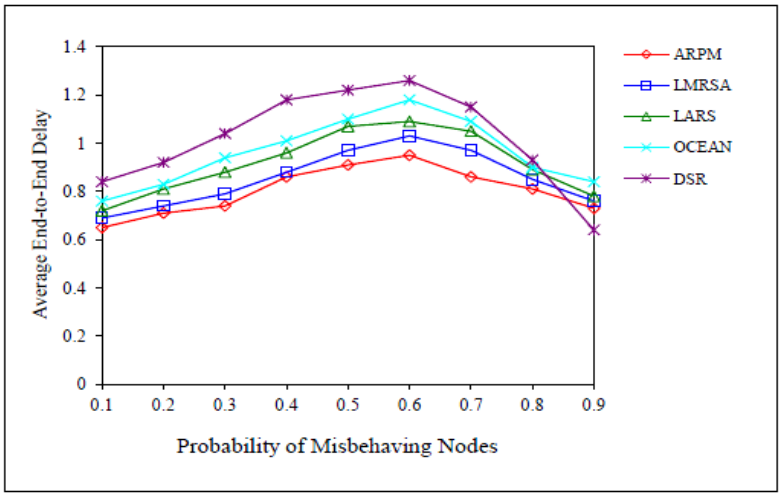


**Fig. 6.** Average end-to-end delay for the proposed scheme and other reputation schemes

ARPM shows minimum End-to-End Delay out of all the deployed schemes as it is capable of efficiently detecting misbehaving nodes. This enables the nodes to find alternate routes avoiding these misbehaving nodes, thereby decreasing the delay. It can be observed that the

delay decreases for high values of $P_M$ (i.e. greater than 0.6). High percentage of misbehaving nodes in the network restricts the communication to within the transmission range of nodes and packets are transferred only between the neighbour nodes, resulting in decreased average end-to-end delay.

**Fig. 7** presents the throughput of the proposed scheme with varying Requite Timer $T_R$.The Probability of Misbehaving Nodes, $P_M$ in the network is 0.3. The throughput is determined over two scenarios having 50 and 100 nodes.

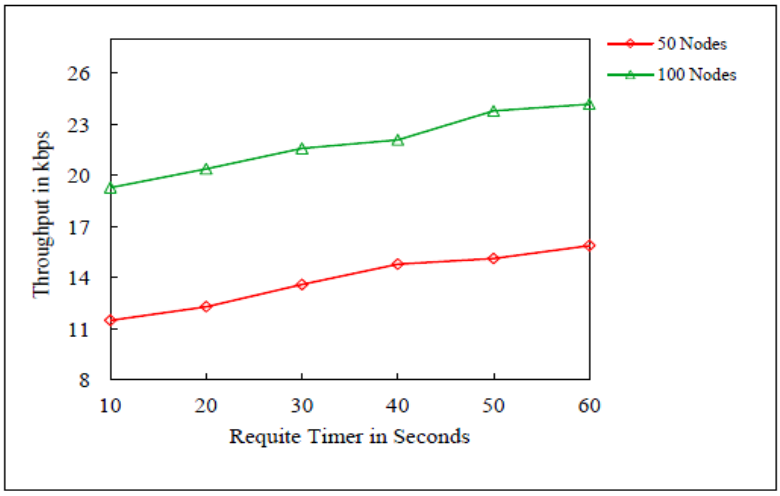


**Fig. 7.** Requite timer $T_R$ of the proposed scheme

Throughput gradually increases as the timer value increases. If $T_R$ is small, then the *Credit Amount* for each node will be calculated quite frequently which may increase the system overhead. Therefore, an optimum value must be chosen. Hence, the Requite timer $T_R$ is set to 30 seconds.

## 5. Conclusion

Limitation of resources in MANETs urges nodes in the network to misbehave. A node may choose to misbehave dynamically to conserve its resources. This paper proposes an Adaptive Security Model which efficiently handles the dynamic behaviour of nodes. It provides an Adaptive Requiting and Punitive Mechanism which is based on the antecedent forwarding behaviour of nodes. The Adaptive Requiting Mechanism encourages the nodes to participate in the network by requiting its good behaviour in the form of credit-chips. The Credit Amount given to nodes depends upon the past communication with that node. Moreover, unlike existing techniques, it gives differential treatment to non-participating nodes and nodes on the periphery of the network. The Adaptive Punitive Mechanism handles packet drop and On/Off Misbehaviour using a time-out based approach. Misbehaving nodes are punished by ostracizing them upon detection which re-enter the network only after a calculated time period. The time period is extended adaptively depending upon its frequency of misbehaviour and the prior communication with that node.

Detailed simulations of the proposed scheme have been performed in network simulator NS2. The results obtained from simulations indicate that the proposed scheme outperforms other reputation based schemes and traditional-DSR protocol in the existence of misbehaving

nodes. Moreover, the simulation outcomes depict that the proposed scheme efficiently handles the dynamic behaviour of nodes. It improves the network performance but at the cost of some computational overhead. Higher computational overhead leads to higher energy consumption which may not be desirable in MANETs. Therefore, an energy efficient adaptive reputation system must be designed, which forms the basis of the future work.

## References


[1] H. Shen and Z. Li, "A Hierarchical Account-Aided Reputation Management System for MANETs," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 70-84, Feb. 2015. Article (CrossRef Link).

[2] T. Eissa, S. Abdul Razak, R. Khokhar and N. Samian, "Trust-Based Routing Mechanism in MANET: Design and Implementation," *Mobile Networks and Applications*, vol.18, no. 5, pp. 666-677, October 2013. Article (CrossRef Link).

[3] G. R Karri and P. M. Khilar, "Routing misbehavior detection and reaction in MANETs," in *Proc. of Proceedings of IEEE International Conference on Industrial and Information Systems*, pp. 80-85, July-August 2010. Article (CrossRef Link).

[4] A. Dhir and J. Sengupta, "Security-aware optimized link routing protocol for mobile ad-hoc networks," *KSII Transactions on Internet and Information Systems*, vol. 3, no.1, pp. 52-83, February 2009. Article (CrossRef Link).

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

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

[7] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta and K.-K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Computer Networks*, March 2017. Article (CrossRef Link).

[8] X. Li, J. Niu, S. Kumari, F. Wu and K.-K. R. Choo, "A robust biometrics based three-factor authentication scheme for Global Mobility Networks in smart city," *Future Generation Computer Systems*, April 2017. Article (CrossRef Link).

[9] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, July 2017. Article (CrossRef Link).

[10] 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 Proceedings of IEEE Global Telecommunications Conference*, pp. 1-6, 30 November- 4 December 2009. Article (CrossRef Link).

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

[12] 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 Proceedings of IEEE International Conference on Recent Trends in Information Technology*, pp. 313-317, 3-5 June 2011. Article (CrossRef Link).

[13] M. Jo, L. Han, D. Kim and H.P. In,"Selfish Attacks and Detection in Cognitive Radio Ad-hoc Networks," *IEEE Network*, vol.27, no.3, pp.46-50, June 2013. Article (CrossRef Link).

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

[15] J. Hu and M. Burmester, "LARS: A locally aware reputation system for mobile ad hoc networks," in *Proc. of Proceedings of the Forty Fourth annual Southeast Regional Conference*, pp. 119-123, 10-12 March 2006. Article (CrossRef Link).
[16] 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 Proceedings of International Conference of Information and Communication Technologies, CCIS*, vol. 101, pp. 344-349, 7-9 September 2010. Article (CrossRef Link).
[17] L. Buttyan and J.P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proc. of First ACM International symposium on Mobile Ad hoc Networking and Computing*, pp. 87-96, 2000. Article (CrossRef Link)
[18] S. Zhong, J. Chen and Y.R. Yang, "Sprite: a simple, cheat-proof credit-based system for mobile ad-hoc networks," in *Proc. of Proceedings 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)
[19] 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)
[20] S. Buchegger and J.Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks)," in *Proc. of MobiHOC 2002: IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing*, pp. 226-236, 2002. Article (CrossRef Link).
[21] P. Michiardi. and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks," in *Proc. of Proceedings of Sixth International Federation for Information Processing Conference on Security Communications and Multimedia Security*, pp. 107-121, 26-27, September, 2002. Article (CrossRef Link).
[22] 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).
[23] T. Chen, F. Wu and S. Zhong, "FITS: A Finite-Time Reputation System for Cooperation in Wireless Ad Hoc Networks," *IEEE Transactions on Computers*, vol. 60, no. 7, pp.1045-1056, July 2011. Article (CrossRef Link).
[24] Z. Noorian, M. Noorian, M. Fleming and S. Marsh, "A Strategic Reputation-Based Mechanism for Mobile Ad Hoc Networks," in *Proc. of Proceedings of Twenty-Fifth Canadian Conference on Artificial Intelligence*, pp. 145-157, 28-30 May 2012. Article (CrossRef Link).
[25] B. Paramasivan, M.J.V. Prakash and M. Kaliappan, "Development of a secure routing protocol using game theory model in mobile ad hoc networks," *Journal of Communications and Networks*, vol. 17, no. 1, pp. 75-83, February 2015. Article (CrossRef Link).
[26] M. A. Azer and N. G. E. D. Saad, "MDAC: A new reputation system for misbehavior detection and control in ad hoc networks," in *Proc. of Proceedings of IEEE International Computer Science and Engineering Conference*, pp. 1-6, 23-26 November 2015. Article (CrossRef Link).
[27] S. Jianhua, S and M. ChuanXiang, "A reputation-based scheme against malicious packet dropping for mobile ad hoc networks," in *Proc. of Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 3, pp. 113-117, 20-22 November 2009. Article (CrossRef Link).
[28] M. T. Refaei, L. A. DaSilva, M. Eltoweissy and T. Nadeem, "Adaptation of Reputation Management Systems to Dynamic Network Conditions in Ad Hoc Networks," *IEEE Transactions on Computers*, vol. 59, no. 5, pp. 707-719, May 2010. Article (CrossRef Link).
[29] X. Gu, P. Lin and S. Shi, "A Novel Reputation Model Based on Subjective Logic for Mobile Ad Hoc Networks," *Advanced Technology in Teaching, Advances in Intelligent and Soft Computing*, vol.163 pp. 525-532, 2013. Article (CrossRef Link).
[30] 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)

[31] X. Mao and J. McNair, "Effect of on/off misbehavior on overhearing based co-operation scheme for MANET," in *Proc. of Proceedings of Military Communications Conference, MILCOM 2010*, pp. 1086-1091, 31 October-3 November 2010. Article (CrossRef Link)
[32] T. Issariyakul and E. Hossain, "Introduction to network simulator NS2," *Springer*, 2nd Ed. USA, 2009. Article (CrossRef Link).

**Dr. Anjali Anand** is working as Lecturer in Computer Science and Engineering Department, Thapar University, Patiala. She has done her M.Tech and Ph.D. from Department of Computer Engineering, Punjabi University, Patiala. She has contributed 7 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 Analysis. 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. Himanshu Aggarwal can be contacted at: himanshu.pup@gmail.com