Avoidance of Malicious Nodes in Mobile Adhoc Networks Using Behavioural Trust Detection and Prevention Framework

Sibomana Fabrice E.J Thomson Fredrik
Department of Computer Science Department of Computer Application
Karpagam Academy of High Education Karpagam Academy of High Education
Tamilnadu, Coimbatore – 21, India Tamilnadu, Coimbatore – 21, India
sibofabrice0308@gmail.com thomson500@gmail.com

Abstract:

Secure routing in Mobile Ad hoc Networks (MANETs) has been a major research since past decade. However, the effects of malicious nodes poses serious threat to data security. Specifically, Wormhole attack is one of most vulnerable attacks, which is hard to detect during routing. Various solutions are developed to counter wormhole attacks, however, at certain extent these techniques tend to fail in detecting the wormhole node and its tunnel. Hence an effective mechanism is required to detect and prevent malicious nodes affected by wormhole attack in MANET. In this paper, a Behavioral Trust Detection and Prevention (BTDP) Framework is proposed to improve the detection and prevention of malicious nodes in MANETs. The framework is responsible for observing the malicious behavior of nodes in network during its mobility and communication between the nodes. It helps to avoid the malicious nodes that affect the packets, which is routed using trust degree level. This detection and prevention helps to improve the routing of packets between nodes in MANET with high data privacy. Experimental results against conventional trust based security protocol prove that the proposed novel BTDPF is efficient in terms of its packet drop rate, false positive rate and wormhole detection time.
Keywords:
MANET, Secure routing, Behavioral Trust Detection, Trust Degree, DPS (Detection and Prevention System)

1. Introduction

Mobile Ad hoc Networks (MANETs) is deployed with massivenetwork infrastructure, sensors and hops. However, the network infrastructure is not of closed one but the protection of data transmission is done in a secured way. The presence of security protocol in network ensures all the operations of network is done at normal rate. During the process of communication and data sharing with other nodes, various problems exist due to attacker interruption [14], poor routing and poor data delivery. In order to avoid these problems, it is possible to use trust based algorithm between the nodes in MANETs.

The presence of wormhole nodes in network attracts increased flow of traffic around its region and it ascertains an abnormal nodes behavior. The operations of wormhole nodes vary slightly from normal nodes i.e. it operates with longer propagation delay, large transmission range, involves in most routes and RREQ message is forwarded to colluding node. The wormhole node transmits lesser packets to neighboring nodes than a normal node. The wormhole node fails to broadcast the RREQ message and a private channel is required for forwarding the RREQ. A tunnel is established between two wormhole nodes to transmit the RREQ packets. During the node movement, the encapsulated packet in tunnel is lost and it is rebroadcasted to the normal nodes with reduced hop count. It greatly affects the transmission of original packet to its respective destination. The routing of packets is affected since there exist a virtual tunnel node by wormhole attacks. Such ineffective routing increases the network overhead and reduces the reliability of
packet delivery in MANETs. Routing process is dynamic due to infrastructure-less routing in MANETs. Moreover, the presence of wormhole changes periodically and it increases with the increasing network dynamicity. The rapid change in tunnel node destroys the network and throughput efficiency in MANETs.

Such effects in MANETs are avoided using a proposed Behavioral Trust Detection and Prevention (BTDP) Framework. It combines trust based detection model and node based prevention model. The framework is designed based on the objective that RREQ packet is transmitted in less by wormhole node than normal nodes. Hence, the proposed method estimates initially the trust between two nodes using direct, indirect or mutual trust. Once, the trust is established between any two nodes, a prevention node helps to broadcasts the authenticity of these two nodes to all the nodes in network. If there is no mutual trust between any two nodes, the node is considered malicious and a threat or block message is broadcasted to all the nodes in network. This prevention node is different from normal nodes. The flow diagram of proposed work is given in Figure 1.

The rest of the paper is organized as follows. Section 2 presents the trust detection model. Section 3 presents the prevention framework to discard the malicious node. Sections 4 discuss the performance analysis of proposed work. Finally, the proposed work is concluded in Section 5.

2. Related works

In [1], Marchand N and data R. 2011 proposed a certain light-weight trust based routing protocol with intrusion detection system to find the trust between two nodes, which mitigates the attacks created by grey hole and black hole.

In [2], Gunasekaran, M., & Premalatha, K. proposed an approach called TEAP: trust-enhance anonymous on demand routing protocol. This approach identifies
and report the nodes that are misbehaving in the network using the concept of anonymity.

In [6], Laxmi V et al proposed a mechanism that analyze the behavior and effect of jellyfish attack on TCP based MANET. Laxmi V and his friend used DTD algorithm to detect and mitigates jellyfish attacks where each node would use locally calculated trust value that were collected within a time period to identify it its neighbor are a jellyfish attacks or not.

In [7], Imran M. et al analyze the danger poses by wormhole attack and the techniques that have been proposed in the previous studies.

In [11], Ahmed N Malik et al proposed a flooding factor based framework for trust management. This flooding approach uses trust values to identify attack nodes in Manet.

In [12], Tiruvakabu, D.S.K and Pallapa, D. proposed an approach that confirms wormhole attack in MANET called Wormhole Attack Confirmation (WAC) System. This approach uses honeypot to eliminate false attack and preserve its resources. The wormhole attack has been a serious threat in MANET than other attacks, hence, it is considered in this work to improve the security in MANETs. Trust-based Source Routing protocol avoids intruders during packet transmission with reduced packet drop and latency.

In [3], Xia, H., Jia, Z., Li, X., Ju, L., & Sha, E. H. M proposed dynamic trust prediction model that evaluate the trustworthiness of nodes in MANET. The trustworthiness will be based on historical and future behavior via an extended fuzzy logic rules prediction.
In [5], Tan, S., Li, X., & Dong, Q proposed trust-based routing mechanism to eliminate malicious node by evaluating trust values of mobile nodes.

In [9], Cho, J. H., Chen, R., & Chan, K. S introduced a composite trust-based public key management with a goal to maximize the network performance while eliminating security risk.

In [4], Wang, B., Chen, X., & Chang, W proposed a Trust-based QoS model that calculates the degree of trust between direct and indirect trust computation. It increases the rate of malicious node detection.

In [8], John, S. P., & Samuel, P. proposed a self-organized key management technique uses normal node and a coordinator node to maintain the security. The present study uses a new trust level called mutual trust, which prominently accepts the mutual trust between two nodes, when the data is transmitted in full duplex model.

In [10], Rajkumar, B., & Narsimha, G. have developed a CA distribution and trust-based threshold revocation method that improve the security of the network with trusted certificate exchange. The proposed study is advanced to a certain extent by considering normal, malicious and multiple prevention or multiple coordinate nodes to detect the wormhole attack.

In [13], Sibomana fabrice and E. J. Thomson Fredrik did a thorough review on recent proposed detection and prevention mechanisms that use node behavioral analysis to detect and prevent malicious node in MANET and concluded by show which ones are more effective than the others.
3. Trust Detection Model

This section discusses the trust relationship between the nodes in network. The trust value between the nodes is considered in terms of direct and indirect trust.

3.1. Calculation of Trust

The trust value in MANET, say $G$ is calculated between nodes $u$ and $v$. Initially, the sensor nodes $u$ and $v$ are justified in terms of its nearby behavior i.e. the nodes are adjacent to each other. If the node $u$ is adjacent to node $v$, then the trust value between nodes $u$ and $v$ is estimated as direct trust model $d(u,v)$. If the node $u$ is not
adjacent with node $v$, then the trust value between nodes $u$ and $v$ is estimated as indirect trust model $i(u,v)$. Finally, the trust $t(u,v)$ between the nodes $u$ and $v$ are figured out. Similarly, the trust $t(v,u)$ between the nodes $v$ and $u$ are calculated in same manner. A comparison between $t(u,v)$ and $t(v,u)$ are made in order to find the mutual trust.

3.1.1. Direct trust

The direct trust model is estimated in terms of communication of nodes, active cooperation of nodes and association of nodes with network to a certain degree, which defines the extent of trust. This direct trust provides relationship between the nodes in terms of its subjective actions, which is an apparent and obvious example of direct trust degree. Here, the analysis of direct trust degree is carried out in detail using similarity and tie strength of nodes. Similarly, analysis of indirect trust degree is carried out using distance between the nodes. The direct trust relation in terms of tie strength is shown in Definition 1 and direct trust relation in terms of similarity between nodes is shown in Definition 2.

**Definition 1:** To an adjacent node pair, the strength of tie between sensor nodes is used to find the trust, which forms the direct trust and its degree is calculated as,

$$d_s(u,v) = \frac{w(u,v)}{w(u)}, \text{where } d_s(u,v) \in (0,1]$$  \hspace{1cm} Eq(1)

where $d(u,v)$ defines the degree of direct trust between the nodes $u$ and $v$. $w(u,v)$ defines the strength between nodes $u$ and $v$. $w(u)$ defines the total tie strength between the node $u$ and neighboring node other than node $v$. $w(u,v)$ is also referred as collaborative or interactive number among the nodes in network.
There always exist a homogeneity among nodes in network i.e. similar nodes are correlated between one another. The similarity of node is estimated by measuring the total shared neighbors between any two nearby sensor nodes. When the similarity of nodes are higher, the neighboring nodes tends to overlap each other at a larger extent. Hence, the present node contributes a very less similarity over a larger number of neighboring node.

**Definition 2:** To an adjacent node pair, the similarity between sensor nodes is used to find the direct similarity trust, which is calculated as,

$$d_s(u,v) = \sum_{t \in N(u) \cap N(v)} (I(t))^{1/4}$$  \hspace{1cm} Eq(2)

where $d_s(u,v)$ defines the degree of direct trust using node similarity. The neighboring sets of node $u$ and node $v$ is given by $N(u)$ and $N(v)$, respectively to estimate the node similarity. $I(t)$ defines the degree of penetration of $t$.

Finally, the direct trust is estimated between two adjacent nodes $u$ and $v$ is given as,

$$d(u,v) = d_s(u,v) + d_s(u,v).$$  \hspace{1cm} Eq(3)

### 3.2. Indirect trust

Indirect trust takes into account the transmission of information between the nodes. The indirect connections exist due to non-adjacent nodes opens up connections via intermediate nodes. This leads to indirect trust between the nonadjacent node, which is estimated using direct trust model between the adjacent nodes. The transmission trust between source and target node takes different form, namely, single and multi-path method. The indirect trust using single path method is given in definition 3 and indirect trust using multi path method is given in definition 4.
**Definition 3:** To a non-adjacent source (u) and target node (v) with a single transmission path between them forms an indirect trust of single path. The approachable paths between the non-adjacent nodes are constructed in terms of intermediate relationship between the nodes u and v. The indirect trust of single path is thus estimated as follows:

\[
i_i(u,v) = \begin{cases} 
  \frac{m d_{\text{max}} - d_{u,v} + 1}{d_{\text{max}}} & \text{if } d_{u,v} \leq d_{\text{max}} \\
  0 & \text{if } d_{u,v} > d_{\text{max}} 
\end{cases}
\]  

Eq(4)

where \( mt = \min(d(u, u_1), d(u_1, u_2), \cdots, d(u_n, v)) \), which forms the intermediate route length between the nodes u and v and \( d_{\text{max}} \) defines the trust transmission with maximum distance. The theory suggests that as the transmission distance increases, the integrity and accuracy of information tends to reduce.

**Definition 4:** To a non-adjacent source (u) and target node (v) with two approachable transmissions path between them forms an indirect trust of multi-path. The indirect trust of multi-paths obtains maximal value after the estimated, which is stated as follows:

\[
i_m(u,v) = \max_{\text{paths}(u,v)} \{i_i(u,v)\}
\]  

Eq(5)

where \( i_m(u,v) \) defines the degree of indirect trust in multi-path between node u and node v. The path set between the node u and node v is given by \( \text{paths}(u,v) \). Hence, the trust degree between the node u and node v in the network G is given as,

\[
t(u,v) = \begin{cases} 
  d(u,v) & \text{if nodes are adjacent} \\
  i_m(u,v) & \text{else} 
\end{cases}
\]  

Eq(6)
3.3. Mutual trust

The trust value is estimated between the nodes using direct trust and indirect trust model. The trust between the nodes pairs is always not similar i.e. \( t(u,v) \neq t(v,u) \) and this is justified as the node with directional property. Additionally, the presence of malicious nodes may not send a response to the node, which has sent a message. This creates an unusual behavior on adjacent sensor nodes i.e. disparity in trust. This has a negative influence over accuracy on trust-based detection. Hence, non-directional model is required to create mutual trust between nodes \( u \) and \( v \) and nodes \( v \) and \( u \).

**Definition 5:** A non-directional reciprocal trust between adjacent nodes \( u \) and \( v \) is called as mutual trust. The mutual trust is computed, when the \( T(u,v) = \{\text{trust}(u,v), \text{trust}(v,u)\} \), which is given as follows:

\[
m(u,v) = \begin{cases} 
\min(T(u,v)) & \text{if } \min(T(u,v)) \geq \chi \\
0 & \text{else}
\end{cases}
\]

Eq(7)

Where \( m(u,v) \) defines the mutual trust between node \( u \) and node \( v \), \( \chi \) defines the degree of trust tolerance to control the minimum allowed level of trust in a network.

The conversion of node trust into mutual trust resolves unusual behavior of nodes and reduces the constraints associated with detecting the trust levels with increased accuracy.

4. Prevention Framework

The proposed trust detection based prevention system has normal, malicious and prevention nodes based on its function.
• **Normal nodes** are found commonly in network that sends transfers data with each other. The *block table* shown in Table 1(a) is intended to discard the malicious nodes. The normal node lists or enters the malicious nodes in *block table* provided by prevention node. It drops data packets, Hello, RREP and RREQ messages from malicious nodes. The Table 1(a) shows that prevention node 54 and 63 announced that node 51 and 52 are malicious, which is added in *block table*.

• **Malicious Nodes** capture RREQ message and broadcast it over entire network. The broadcast made by malicious nodes does not increase the hop count. The RREP message is sent again over the same path, where it gets involved into other supplementary routes. Now the source nodes think that routes through these nodes are short and hence it establishes communication via these paths.

• **Prevention Nodes** detect the malicious nodes and block it. These nodes have an analysis table (Table 1b) with a status field that defines the range of prevention node. The nodes lying within the range is set active (nodes 31 and 43) and nodes lying outside the range is set inactive (node 41). The field *RREQ count* shows that nodes broadcasted number of RREQ messages, 6, 5 and 6 for the nodes 31, 41 and 43, respectively. The field *Suspicious Value* represents the suspicious value estimated by nodes. The field *Threat* and field *SuspiciousNodeConfirmed* shows the threat or block message broadcasted against malicious nodes. Finally, the block message (Table 1c) and threat message (Table 1d) is given in the block and threat message table. As per the Table 1, it is seen that node 41 is considered as malicious nodes.
Table 1: Prevention Node Table

Table 1 (a) Block Table

<table>
<thead>
<tr>
<th>Malicious node</th>
<th>DPS node</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>54</td>
</tr>
<tr>
<td>51</td>
<td>63</td>
</tr>
</tbody>
</table>

Table 1 (b) Analysis Table

<table>
<thead>
<tr>
<th>Status</th>
<th>Node ID</th>
<th>RREQ Count</th>
<th>Suspicious Value</th>
<th>Wormhole Threat</th>
<th>Wormhole Confirmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>31</td>
<td>6</td>
<td>0</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inactive</td>
<td>41</td>
<td>5</td>
<td>3</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Active</td>
<td>43</td>
<td>6</td>
<td>0</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1 (c) Threat Message

<table>
<thead>
<tr>
<th>Threat Message</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious Node</td>
<td>50</td>
</tr>
<tr>
<td>Announcer DPS</td>
<td>54</td>
</tr>
</tbody>
</table>
Table 1 (d) Block Message

<table>
<thead>
<tr>
<th>Block Message</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious Node</td>
<td>50</td>
</tr>
<tr>
<td>Announcer DPS Node</td>
<td>54</td>
</tr>
</tbody>
</table>

The total number of prevention nodes depends entirely on transmission range and network area. The analysis is carried out by deploying the prevention node over entire network and it communicates directly with other nodes. The estimation of total prevention nodes is given by,

\[
\text{Prevention Node} = \left( \frac{X}{r} - 1 \right) \left( \frac{Y}{r} - 1 \right)
\]

Eq(8)

where, \(X\) defines the length of network, \(Y\) defines the width of network and \(r\) denotes the range of transmission by prevention node. The prevention nodes has four parameters for various purpose, which is defined below:

**Max_RC:** The suspicious value calculated is initiated for each node, when the RREQ message count of each node reaches the Max_RC in analysis table.

**Min_RC:** After the initial calculation of the suspicious value, a prevention node will check two things. Firstly, the prevention node will check whether RREQ count is lesser than Min_RC or not and secondly, it will check whether the node is in active status or not. The analysis table increments the suspicious value of a node. The Min_RC value ≤ half of Max_RC value.

**Min_TV:** If suspicious value = Min_TV, a threat message is issued by prevention node to alert other prevention nodes. The Min_TV value = half of Max_TV value.
**Max_TV**: If *suspicious value* = *Max_TV*, a block message is issued by prevention node to alert other prevention and normal nodes.

The prevention nodes have four operations, which are given in following algorithms. The Figure 2 shows the count of RREQ messages using a Counting Algorithm. The Figure 3 shows the algorithm that calculates the Suspicious Value. The Figure 4 shows the algorithm to broadcast the Threat Message. Figure 5a shows the block message received by prevention node and Figure 5b block message received by normal node.

### 4.1. RREQ Counting Algorithm

```
Algorithm 1: Neighboring node broadcasts a RREQ message to prevention node
if node.id ∈ analysis_table then
    if node.wormhole_confirmed == yes then
        return
    end if
    if node.status == inactive then
        node.status ← inactive
    end if
    node.rreq_count ← node.rreq_count + 1;
    if node.rreq_count == max_RC then
        calculate suspicious_value();
    else
        return
    end if
else
    create entry(); for neighboring node
    node.status ← active;
    node.rreq_count ← 1;
    node.suspicious_value ← 0;
    node.malicious_threat ← No;
    node.malicious_confirmed ← No;
end if
```

Figure 2: RREQ Counting Algorithm
4.2. Suspicious Value Calculation Algorithm

Algorithm 2: RREQ message count reaches $Max_{RC}$

For all node.$analysis_table$ do
    if node.rreq_count $< Max_{RC}$ & node.status == active then
        node.suspicious_value $\leftarrow$ node.suspicious_value + 1;
        if node.suspicious_value $< max_{TV}$ & node.suspicious_threat == No then
            node.suspicious_threat $\leftarrow$ yes;
            threat_message();
        end if
        if node.suspicious_value $< max_{TV}$ & node.suspicious_confirmed == No then
            node.suspicious_confirmed $\leftarrow$ yes;
            block_message();
        end if
    end if
    else if node.rreq_count $> Max_{RC}$ & node.status == active && node.suspicious_value $> 0$ then
        node.suspicious_value $\leftarrow$ node.suspicious_value $- 1$;
    end if
    node.status $=$ inactive;
    node.rreq_count $=$ 0;
end for
return

Figure 3: Suspicious Value Calculation Algorithm

4.3. Threat Message Broadcasting Algorithm

Algorithm 3: Prevention node receives a threat message

if node.id $\in$ $analysis_table$ then
    if node.wormhole_threat $==$ yes then
        return
    else
        create entry(); for neighboring node
        node.suspicious_value $\leftarrow$ min_{TV}
        node.malicious_threat $\leftarrow$ yes;
        threat_message();
    end if
end if

Figure 4: Threat Message Broadcasting Algorithm
4.4. Block Message Broadcasting Algorithm

Algorithm 4: Prevention node receives a block message

if node.id ∈ analysis_table then
    if node.wormhole_confirmed == yes then
        return
    else
        node.suspicious_value ← max_TV;
        node.malicious_threat ← yes;
        node.wormhole_confirmed ← yes;
        block_message();
    else if
        create entry(); for neighboring node
        node.suspicious_value ← max_TV;
        node.malicious_threat ← yes;
        node.malicious_confirmed ← yes;
        block_message();
    end if
end if
return

Algorithm 5: Normal node receives a block message

if node.id ∈ block_table then
    return
else
    create entry(); for neighboring node
    update_routes();
end if
return

Figure 5(a) Block message received by prevention node

Figure 5(b) Block message received by normal node

Figure 5: Block Message Broadcasting Algorithm

Before the implementation of the above algorithms, the estimation of trust degree between two nodes is carried out and stored statistically in local file system to attain improved performance in detection process using behavioral description.
### Table 2: Simulation parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>1500×1500m$^2$</td>
</tr>
<tr>
<td>Time</td>
<td>500 s</td>
</tr>
<tr>
<td>Protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Nodes</td>
<td>Normal: 50</td>
</tr>
<tr>
<td></td>
<td>Malicious: 2, 8 (few are fixed and few are mobile in 8 wormhole nodes)</td>
</tr>
<tr>
<td></td>
<td>Prevention node: 18 (all are fixed)</td>
</tr>
<tr>
<td>Min$_{RC}$</td>
<td>3</td>
</tr>
<tr>
<td>Max$_{RC}$</td>
<td>7</td>
</tr>
<tr>
<td>Min$_{TV}$</td>
<td>5</td>
</tr>
<tr>
<td>Max$_{TV}$</td>
<td>10</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250 m</td>
</tr>
<tr>
<td>Mobility</td>
<td>Random mobility, 0 – 25 ms$^{-1}$</td>
</tr>
<tr>
<td>Maximum number of connections</td>
<td>50 nodes or 25 pairs</td>
</tr>
<tr>
<td>Type of Traffic</td>
<td>CBR</td>
</tr>
<tr>
<td>Size of data packet</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Maximum speed of packets</td>
<td>25 ms$^{-1}$</td>
</tr>
<tr>
<td>Pause time</td>
<td>0 -20 s</td>
</tr>
</tbody>
</table>
5. Experimental Results and Analysis

The trust calculation and wormhole detection is taken care in E6700 3.2 GHz 4GB RAM. The trust estimation is programmed using R with iGraph toolkit and wormhole detection is carried out using NS-2 (Network Simulator 2) version 2.34. The parametric values required to operate the proposed experiments are given in Table 2.

The performance of proposed system is experimented with 18 prevention nodes and 50 normal nodes in a fixed location using AODV protocol. The Figure 6a presents the network nodes without wormhole nodes and Figure 6b presents the network nodes with 8 wormhole nodes. In Figure 6b, a virtual tunnel is thus created between two wormhole nodes encapsulates RREQ message and forwards it to other malicious nodes as a data packet. The RREQ packet is extracted at other malicious node and further it is rebroadcasted by updating the path at regular intervals. The RREQ packet has lesser hop counts than other nodes and it helps the wormhole nodes to involve in other paths. It further makes the other nodes to drop the data packets which they receive from source node.

The Figure 6b has two major scenarios that include: fixed and mobile wormhole nodes. It is easy to detect the wormhole nodes in the network, which are placed fixed in a specific location. Since, monitoring the behavior of malicious nodes is easier even by normal nodes until the nodes are broadcasted as malicious. On other hand, it is difficult to detect the malicious nodes in network, when the nodes are in mobile mode. Since the malicious nodes go away from the range of monitoring node and enter into the range of other monitoring nodes. Hence, the data collected
by first monitoring node is considered useless. To avoid such effects, the proposed method uses prevention node to share the malicious node information with other nodes using a threat message.

Figure 6(a) MANET with two malicious nodes in original position (red and blue box with solid line) and then in moved position (red and blue box with dotted line); prevention nodes are shown in red colored node.
Figure 6(b) MANET with eight malicious nodes in original position (red and blue box with solid line) and then in moved position (red and blue box with dotted line); prevention nodes are shown in red colored node.

Figure 6: Simulation Scenario

The proposed study considered two major cases: Case 1 with 0 wormhole nodes and Case 2 with 8 wormhole nodes, shown in Figure 6. Both Case 1 and Case 2 is tested on two pause time (0 – 20s) with the execution of multiple times simulation. The average values of packets dropped and the detection time, false positive rate are noted down. The malicious nodes are allowed to move over a new location after 50 seconds in simulation scenario. The Figure 6b shows the current node location as solid rectangle and new node location as dotted rectangle.
1.1. Packet Drop rate:

The Figure 7a shows the packet drop rate between proposed BTDP with AODV protocol and existing honeypot for 0 and 20 pause time with two fixed wormhole and mobile wormhole nodes. The average packet drop rate for Honeypot is 17.6% and 19.3%, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average packet drop rate for BTDP is 12.5% and 12.6%, respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced packet drop rate of 28.97% and 34.7% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes. The Figure 7b shows the packet drop rate between proposed BTDP with AODV protocol and existing Honeypot for 0 and 20 pause time with eight fixed wormhole and mobile wormhole nodes. The average packet drop rate for Honeypot is 25.9% and 28.4%, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average packet drop rate for BTDP is 13.4% and 13.1%, respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced packet drop rate of 48.26% and 53.87% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes.

Figure 7(a)
1.2. False Positive Rate

The Figure 8a shows the false positive rate difference between proposed BTDP with AODV protocol and existing Honeypot for 0 and 20 pause time with two fixed wormhole and mobile wormhole nodes. The average false positive rate for Honeypot is 6.4% and 5.8%, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average false positive rate for BTDP is 0.2% and 0.4%, respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced false positive rate of 96.87% and 93.10% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes. The Figure 8b shows the false positive rate difference between proposed BTDP with AODV protocol and existing Honeypot for 0 and 20 pause time with eight fixed wormhole and mobile wormhole nodes. The average false positive rate for Honeypot is 9.7% and 12.5%, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average false positive rate for BTDP is 0.4% and
0.1%, respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced false positive rate of 95.8% and 99.2% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes.

Figure 8: False Positive Rate between rate between BTDP and HONEYPOT
1.3. Wormhole Detection Time

The Figure 9a shows the detection time difference between proposed BTDP with AODV protocol and existing Honeypot for 0 and 20 pause time with two fixed wormhole and mobile wormhole nodes. The average detection time for Honeypot is 200s and 248s, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average detection time for BTDP is 124s and 122s respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced detection time of 38% and 49% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes.

The Figure 9b shows the detection time difference between proposed BTDP with AODV protocol and existing Honeypot for 0 and 20 pause time with eight fixed wormhole and mobile wormhole nodes. The average detection time for Honeypot is 300s and 354s, respectively for fixed wormhole and mobile wormhole nodes. Similarly, the average detection time for BTDP is 149s and 169s, respectively for fixed wormhole and mobile wormhole nodes. Hence, it is seen that proposed system obtains a reduced detection time of 50.3% and 52% than Honeypot, respectively for fixed wormhole and mobile wormhole nodes.

![Figure 9(a)](image-url)
Figure 9(b)

Figure 9: Wormhole detection time between BTDP and HONEYPOT

6. Conclusion

In this paper, we present a trust based detection and prevention model in Mobile Ad hoc Networks against wormhole attack. This method is intended to check the trust level of each node and further it prevents the wormhole network in disrupting the packet flow. The prevention model reduces the number of RREQ packets being broadcasted to neighboring nodes. Since normal nodes do find accurately the wormhole node, we then use a prevention node in the network to find the malicious nodes. This node involves in finding and eliminating the RREQ broadcast message, but not in normal routing process. The result shows that proposed BTDP has a higher detection rate against wormhole nodes. The proposed BTDP method prevents the normal node being affected by wormhole behavior and it reduces the detection time than Honeypot model. The proposed method also eliminates the spreading of false information throughout the MANET and extends life duration of nodes.
References


