A Trust Based Secure Source Routing using Fuzzy Logic Rules Prediction in Mobile Ad Hoc Networks

A. Duraimurugan, S. Karthi, D. Kirupagaran

Abstract—A Mobile Ad hoc Network (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links. It is a collection of independent mobile nodes that can communicate with each other via radio waves. These networks are fully distributed, and can work at any place without the help of any infrastructure. MANETs are much more susceptible to attack than the wired network. Due to the dynamically changing network topology, mobile nodes often comes in and goes out of the network, thereby allowing any malicious node to join the network without being detected. Hence Ad Hoc network needs very specialized security methods. But there is no single approach fitting all the networks, as the nodes can be any devices. Therefore, in this paper, we propose an Enhanced Trust-based secure source routing (ETSR) protocol using Fuzzy Logic Rules Prediction, which provides a malleable and defeasible approach to choose the shortest route that meets the security requirements of data packet transmission with an adaptive trust level classification of nodes. This is done by bearing in mind the average trust level of all the nodes. We have also analysed the parameters such as packet delivery ratio, routing packet overhead, network throughput, and average end-to-end latency, in comparison with the existing Trust-based source routing (TSR) protocol.


I. INTRODUCTION

An ad hoc network is a collection of mobile nodes forming a temporary network without the aid of any centralized administration or standard support services regularly available on conventional networks. The increased popularity and usage of wireless technologies has opened the doors for new emerging applications in the domain of networking. One emerging and promising areas is the domain of Mobile Ad Hoc Networks (MANETs). A mobile ad hoc network is a collection of wireless mobile nodes that form a dynamic network without the need for infrastructure or centralized points. The dynamic nature of ad hoc networks presents many security challenges. Secure routing is a promising area for achieving better security for the network by protecting the routing protocols against malicious attacks. Several secure routing protocols have been proposed in the literatures that were successful in avoiding and preventing some types of security attacks in MANETs. However, MANETs are still vulnerable to other types of attacks. Hence, there is a need for introducing an efficient mechanism to detect malicious nodes. Dynamic changes in network's topology causes weak trust relationship among the nodes in the network. In MANETs a mobile node operates as not only end terminal but also as an intermediate router. Therefore, a multi-hop scenario occurs or communication in MANETs; where there may be one or more malicious nodes in between source and destination. A routing protocol is said to be secure that detects the detrimental effects of malicious node(s in the path from source to destination). In this paper, a new mechanism is proposed that improves the performance of routing protocol against the malicious attacks. Since a malicious node behaves in abnormal ways, this mechanism proposes observing nodes behavior such as nodes’ mobility, and avoiding communication through these nodes which may lead to more secure routing.

II. RELATED WORK

In reference [12], Hughes et al. propose Dynamic Trust-based Resources (DyTR), which uses trust evaluation as a method of access control to network resources. The authors, however, do not discuss the securing of the trust information exchange. Pirzada and McDonald develop a protocol based on DSR in [13]. Their protocol takes advantage of the full route information available in DSR. Unlike other recommendations, however, they only consider trust from direct observations rather than including third party opinions. In their protocol, however, lazy nodes are not penalized and there- fore have no incentive to participate. Pirzada et al. [14] evaluated the performance of three trust-based reactive routing protocols (trusted AODV, DSR and TORA) by varying the number of malicious nodes and other experiment settings. The results indicate that each trust-based routing protocol has its own advantage. In particular, trust-based AODV routing maintains a stable throughput and surpasses TORA and DSR at higher traffic loads.

III. EXISTING SYSTEM

A. Trust-based source routing protocol

Trust-based Source Routing protocol (TSR) is a novel on-demand trust-based unicast routing protocol for MANETs, which provides a flexible and feasible approach to choose the shortest route that meets the security requirement of data packets transmission [1]. TSR is a source routing (on-demand routing) protocol that integrates a dynamic trust prediction model to evaluate the trustworthiness of nodes, which is based on the nodes’ historical behaviours, as well as the future behaviours via extended fuzzy logic rules prediction. In this
trust model, passive acknowledgment (i.e., monitored node’s packet forwarding ratio) is used as the single observable factor for assessing trust. Passive acknowledgment uses promiscuous mode to monitor neighbours’ behaviours in the wireless radio channel, which allows a node to detect any transmitted packet in its communication range, irrelevant of their destinations.

B. Limitations of the Existing System

i. This system lacks an adaptive trust level classification of nodes. Adaptive trust level classification of nodes provide a highly accurate trust value for the nodes.

ii. This system faces routing cache problem, which leads to network overhead. Routing cache stores the recent routes that have been used for data transmission for future use. But according to this system, it ignores this cache as the trust requirement is not satisfied. So every time it initiates a new routing process which is an overhead in this type of system.

III. PROPOSED SYSTEM

Our proposed work, Enhanced Trust-based secured source routing using fuzzy logic rules prediction (ETSR) is the augmentation of the Trust-based source routing protocol (TSR), by adapting the trust model which was proposed in the existing system. Our proposed work overcomes the limitations of the Trust-based source routing protocol and proves to perform better than the existing system. In our proposed work, we have included the adaptive trust level classification of nodes by considering both the individual trust values of each node and also the average trust value of all the nodes in the trusted network. Adaptive Trust Level Classification provides the required security by without degrading the performance of the system and with no overheads to the system. By way of considering the average trust values of the intermediate nodes to compute the route’s trust, our proposed work performs better than the TSR.

A. Architecture

The architecture of our proposed system, consists of four major blocks namely,

i. Route Discovery

ii. Trust Computation

iii. Trust Application

iv. Route Maintenance

The detailed description about the architecture is as follows.

![Fig. 1. Overall Architecture of the Proposed System](image)

Figure 1 shows the overall architecture of the proposed work. Route discovery block is nothing but the source routing protocol. Source routing protocol is creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by a route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer desired. Trust computation block involves the process of computation of the node’s trust value based on the adopted trust model. The node’s trusts such as the node’s historical trust values and the node’s current trust values are computed in this block. The node’s historical trust value is computed by observing the node’s behaviour. Based on the historical trust value computed and the node’s capability level, the node’s current trust is computed by using the fuzzy logic rules prediction. This two trust value is of great importance and plays a major role in establishing a trusted route to the destination. Trust application block does the process of integrating the trust computation with the source routing protocol. In this block, the source routing protocol makes use of the computed node’s trust values to establish a more reliable and trusted path to the destination. The trusted path is established by the source node to the destination node by satisfying the trust requirement of the system that is nothing but the black-list trust threshold (η). Route maintenance block involves the process of route update. Route update is the process of updating the route from source to destination during the process of data transmission. Due to the mobility nature of the nodes, there is a risk of broken links and alternative routes from the source to destination. Hence route maintenance plays an important role in the process of the trusted source routing protocol.

B. Module Design

Figure 2 shows the module design of our proposed system. It comprises of two main modules namely,

i. Trust Computation

ii. Trust Application

![Fig. 2. Module Design for the Proposed System](image)

The two main modules, in turn, consists of two sub-modules. They are:

a. Computation of Node’s Trust

b. Computation of Route’s Trust

c. Trusted Route Discovery
d. Punishing Malicious Nodes

i. Trust Computation: In ad hoc networks, ‘trust’ is a relationship between two neighbour entities. We define trust in this model as the reliability, timeliness, and integrity of message delivery to their intended next-hop. Simply speaking, trust expresses the degree that one node expects another node to offer certain services. Our proposed framework evaluates trust on a continuous scale and takes into account both node trust and route trust. In our model, a node’s trust can be considered as a subjective measurement of the node’s quality of forwarding, while a route’s trust can be used to anticipate the quality of forwarding packets along the route. Overall, trust model essentially performs trust derivation, computation and application [9].

Observing a node’s behaviours is an effective mechanism to determine whether this node can be trusted. Meanwhile, we find that the conjunction of subjective passive acknowledgment and node’s capability level on providing services can give an effective indication of a node’s behaviour of cooperation. We assume each node should be able to detect behaviour exhibited by malicious nodes indicative of Sybil attacks. Distributed Intrusion Detection Systems (IDSs) can fulfill this requirement as every node possesses an IDS and can detect attacks from other nodes in its range [18],[5]. Threat information from the IDSs will be communicated using standard data packets, requiring no special control transmissions. We also assume normal nodes are stimulated to cooperate adequately on the network. Because cooperation is naturally required for a functional MANET to exist, acceptable behaviour and participation in the network must be encouraged [15]. Basing on the above assumptions, we design our trust model as follows. Initially each node in the system is authenticated by an authentication mechanism if possible. We consider the interactions which occur between a node and its single-hop physical neighbour as direct interactions. In order to simplify trust model, we only use the history of direct interactions among nodes to compute ‘trust’. In our trust model, passive acknowledgment (i.e., monitored node’s packet forwarding ratio) is used as the single observable factor for assessing trust. Passive acknowledgment uses promiscuous mode to monitor neighbours’ behaviours in the wireless radio channel, which allows a node to detect any transmitted packet in its communication range, irrelevant of their destinations. In this model, at time t, TV(t) denotes for a node’s trust, which is defined in a continuous range between 0 and 1 (i.e., 0 ≤ TV(t) ≤ 1). Let vi and vj represent the evaluating (monitoring) and evaluated (monitored) nodes, respectively. The trust value 0 signifies complete distrust, while the value 1 implies absolute trust. In our model, there are three types of trust, which are historical trust, current trust and route trust [4],[2].

a. Computation of Node’s Trust

As mentioned in the previous section node’s trust is classified into two such the node’s historical trust and the node’s current trust. Their computation processes are described in detail as follows.

i. Computation of node’s historical trust: We assume that, after one node broadcasting a packet, all its neighbours will receive the packet correctly. However, if the distance between source and destination is beyond one hop, packet might be dropped by intermediate nodes due to unexpected causes (e.g., heavy traffic or selfish node’s case) or malicious attacks (e.g., blackhole or grayhole attacks). Trust evaluation in a routing procedure is an assessment of forwarding behaviours of neighbours by a sender. More specifically, a node vi will give its neighbour vj, a trust score after the node vi transmits a packet sent by node vj [4],[7]. Thus we use packet forwarding ratio to evaluate the quality of forwarding.

Forwarding Ratio (FR): It is the proportion of the number of packets forwarded correctly to the number of those supposed to be forwarded. Correct forwarding means a forwarding node not only transmits a packet to its next hop node but also forwards devotedly. For instance, when a malicious neighbour node forwards a data packet after tampering with data, it is not considered as correct forwarding. If the sender monitors this illegal modification, the forwarding ratio of this neighbour will decrease. At time t, FR(t) is computed as follows:

$$\text{FR}(t) = \frac{N_{\text{corr}}(t)}{N_{\text{all}}(t)}$$

where,

- $N_{\text{corr}}(t)$ Represents the cumulative count of correct forwarding packets and,
- $N_{\text{all}}(t)$ Signifies the total count of all requesting packets from time 0 to t.

In mobile ad hoc networks, all packets can be classified into two types: control packets and data packets. The accuracy of control packets plays a vital role in establishment of accurate routes in the network. So FR is divided into two parts: Control packet Forwarding Ratio, denoted by CFR, and Data packet Forwarding Ratio, denoted by DFR. They are computed using forwarding count of control packets and data packets according to formula (1) respectively. Trust record list is introduced to record the CFR and DFR trust information. Each node will maintain a trust record list for every one-hop neighbour to which packets have been sent for forwarding. A trust record list showed in Table 1, it contains monitored node ID, node’s historical trust value, two integer counters of $N_{\text{corr}}$, and $N_{\text{all}}$ for control packets, two integer counters of $N_{\text{corr}}$ and $N_{\text{all}}$ for data packets, and a packet buffer. The packet buffer is used to record all packets sent recently. The specific approach is: before sending a packet, a sender increases $N_{\text{corr}}$ for control packets or $N_{\text{all}}$ for data packets by 1. For a broadcast packet including a route request packet or a route update packet, the sender increases $N_{\text{all}}$ for control packets of all records in its trust record list except $N_{\text{all}}$ for control packets of the node where the packet comes from. For a
unicast packet, the sender only adds 1 to $N_{all}$ for control packets of the next-hop when the packet is a route reply packet, or adds 1 to $N_{all}$ for data packets of the next-hop when the packet is a data packet. To detect whether a packet is successfully forwarded, the sender will not delete the packet immediately after sent out. The packet will be stored in the packet buffer and wait for acknowledgment. A retry counter RetryCnt is used to remember retransmission number of every packet. If the sender in the promiscuous mode monitors the packet is forwarded correctly, it will be removed from the packet buffer and the corresponding counter of correct forwarding ($N_{corr}$) is increased by 1. 

The trust value of a node $v_i$ in the monitoring node $v_j$ is a measure to ensure that packets sent by node $v_i$ have actually been forwarded by node $v_j$. Two trust factors (CFR and DFR) are assigned weights in order to determine a monitored node’s trust. At time $t$, the historical trust value of node $v_i$ evaluated by node $v_j$ (i.e., represented as $TV_{ij}(t)$) is calculated by the following formula:

$$TV_{ij}(t) = w_1 \times CFR_{ij}(t) + w_2 \times DFR_{ij}(t)$$

Where, $CFR_{ij}(t)$ and $DFR_{ij}(t)$ represent control packet forwarding ratio and data packet forwarding ratio respectively observed by node $v_j$ for forwarding node $v_i$, and the weights $w_1$ and $w_2$ ($w_1, w_2 > 0$ and $w_1 + w_2 = 1$) are assigned to $CFR_{ij}(t)$ and $DFR_{ij}(t)$ respectively at time $t$.

<table>
<thead>
<tr>
<th>Node_ID</th>
<th>Node’s historical trust value</th>
<th>$N_{corr}$ and $N_{all}$ for control packets</th>
<th>$N_{corr}$ and $N_{all}$ for data packets</th>
<th>Packet buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitored node ID</td>
<td>TV(t)</td>
<td>VL</td>
<td>L</td>
<td>M</td>
</tr>
</tbody>
</table>

The rules in the above table actually establish a mapping function from $TV(t) \times C(t)$ to $TV(t + 1)$, which is based on the analysis of the node’s historical behaviors and current conditions. For example, when an overloaded node lacks the CPU cycles, buffer space or available network bandwidth to forward packets, such a low capability level, even if its historical trust level is very high, it will also be considered unreliable in the next time. This only shows the second rule from above table. Corresponding with each rule, there is an inference relationship $R^*_j$:

$$R^*_j = TV(t) \times C(t) \times TV(t+1)$$

That is for $\forall h \in TV(t)$, $C(t), u \in TV(t+1)$, we have $R^*_j(h, c, u) = TV(h)^c C(c)^u TV(u)$. For all the $n$ rules we have the fuzzy inference relationship,

$$R^*_j(h, c, u) = \bigvee_{t \in TV(t+1)} R^*_j(h, c, u)$$

For each pair of given $(TV(t), C(t))$, using the general total relationship $R$, we can obtain an output:

$$TV(t + 1) = TV(t) \times C(t) \times TV(t+1) \circ R$$

Then with the help of the maximum membership degree approach, we can get an explicitly node’s current trust $u^* \in [0, 1]$ by defuzzification. We can recycle the method to update this node’s trust. Finally, each node additionally owns a trust table (Table 3) with items defined as follows:

<table>
<thead>
<tr>
<th>Node_ID</th>
<th>Neighbour_ID</th>
<th>TV</th>
<th>Black-List</th>
</tr>
</thead>
</table>

function of $C(t)$ also consists of four fuzzy sets: VeryLow(VL-cannot afford to provide services), Low(L-low capability level), Medial(M-medium capability level) and High(H-high capability level), respectively. Combined with social control theory, we give the fuzzy inference rules as follows (Table 2):

<table>
<thead>
<tr>
<th>Table 2 Logical rules prediction on trust levels</th>
<th>C(t)</th>
<th>TV(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL</td>
<td>VL</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>VL</td>
<td>VL</td>
</tr>
<tr>
<td>M</td>
<td>VL</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>VL</td>
<td>L</td>
</tr>
</tbody>
</table>

(ii) Computation of node’s current trust: Risk is defined as the uncertainty of loss, the consequences of uncertainty and the negative deviation from the expected target. If the trend of events can be predicted accurately, the probability of risk occurrence can be reduced. In this subsection, considering a monitored node’s historical trust and its current capability to provide services, we use the fuzzy logic rules prediction method to compute this node’s current trust, which can offer an accurate prediction of node’s future behaviors [4],[2]. The evaluated node’s historical interactions can be recorded and its capability level can be monitored, therefore, we can model the two factors as follows: let $TV(t)$ represents for a monitored node’s historical trust level at time $t$, its record of history behaviors on offering certain services in the past few time intervals, just like packet forwarding ratio or packet-drop ratio, which has been measured; let $C(t)$ represents for the node’s capability level on providing packets transmission services at time $t$, which includes the remnant utilization ratio of battery, local memory, CPU cycle, and bandwidth at that point; let $TV(t + 1)$ refers to the node’s trust level at time $t + 1$.

Assume the fuzzy membership function of $TV(t)$ or $TV(t + 1)$ consists of four fuzzy sets: VeryLow(VL-malicious node), Low(L-low trustworthy node), Medial(M-trustworthy node) and High(H-complete trustworthy node), and the fuzzy member...
b. Computation of Route’s Trust

The route’s trust value is calculated as the average trust values of the intermediate nodes between the source and the destination. It is denoted by RouteTV_{sd}(t). It is calculated as:

$$ RouteTV_{sd}(t) = \frac{\sum_{i=1}^{n}[TV_{ij}(t) + TV_{ji}(t) \text{ and } v_i \rightarrow v_j]}{n} \ldots (7) $$

where, $v_i$ – represents the source node of route P, $v_d$ – represents the destination node of route P, $TV_{ij}(t)$ – represents the Node’s trust value, $v_i$ and $v_j$ are any two adjacent nodes along the route P, $v_i \rightarrow v_j$ means that $v_j$ is the next-hop node of $v_i$ and $n$ – represents the number of intermediate nodes.

The computation of route trust takes into account trust values of all nodes in the route excluding the destination node. This is because we do not consider the trust values evaluated by the destination node. The computation of route trust complies with the opinion in information theory: the information cannot be increased via propagation [10]. Moreover, in our proposed work, route trust is the trust experienced by the last packet which has arrived along the route. Since network load conditions will change from time to time during the connection, the trust will also change accordingly. By using the latest arrived data packet to calculate RouteTV_{sd}(t), the scheme is adaptive to changing network conditions and the source will be correctly informed in a timely manner for a ‘Handoff’ so the packet losses can be minimized. In the example below, the Route Trust for the path (A→B→D→F) is calculated as 0.93 by using the equation(7).

Fig. 3. Computation of Route Trust

ii. Trust Application. A node’s trust value not only provides a relative identification between the normal node and the malicious node, but also offers a prediction of this node’s future behaviours. We present a simple case to illustrate the trust application. The service node s determines route trust requirements basing on the service level. We can simply divide the service in a file sharing system into three levels: important documents sharing (I), the less important documents sharing (L), and the regular documents sharing (R). For those shared files, we define a mapping function f (as shown in Eq. (8)).

$$ f(RouteTV_{sd}) = \begin{cases} 1 & 0.9 \leq RouteTV_{sd} \leq 1 \\ 0.8 & 0.8 \leq RouteTV_{sd} \leq 0.9 \\ 0.7 & 0.7 \leq RouteTV_{sd} \leq 0.8 \end{cases} \ldots (8) $$

We can also set different boundary values (e.g., $0.95 \leq RouteTV_{sd} \leq 1$) corresponding with the special security needs of the network. According to this file sharing system, we define a simple grading criteria for node’s trust levels, which is shown in Table 4. The trust threshold values in Table 4 (e.g., 0.7) are just for illustration purpose. In practice, they will be determined based on the characteristics and safety requirement of the real application, and given as input parameters to our evaluation model. Each node, based upon its personal experiences, rewards collaborative nodes for their beneficent behaviours and punishes malicious nodes for their malevolent actions.

l (i) Route discovery and path selection

When a sender (or source) needs to send a flow to the destination, it will firstly look up its routing cache to see if it has an unexpired qualified trusted route to the destination. If it does, it will send data packets using the route immediately. Otherwise it broadcasts a FLOW-REQ message to find a route to the destination. This FLOW-REQ message is processed similarly to the ROUTE REQUEST message in DSR [19]. The message contains a sequence number, source ID, destination ID, the list of node IDs and TVs appended by the intermediate nodes while the message travels through the network. A neighbor, upon receiving this message, will forward it to all of its neighbors if : (a) the message contains a higher sequence number for any of the previously received source and destination ID pair, or (b) the message contains the same sequence number but the message arrives from a better route (e.g., the route has a larger RouteTV with the same or fewer number of hops, under the constraint of the qualified route’s trust route ‘RouteTV’ is larger than the requirement of data packet transmission.). When a node is forwarding the FLOW-REQ message, it also appends its own ID and the TV obtained from the evaluating node (or upstream node on the route) that the message received from using a higher layer. When a FLOW-REQ message arrives at the destination, it contains the list of nodes along the route it has travelled and the TVs for each hop along the route. During the connection, intermediate nodes append TVs to each data packet, so the destination continues to receive RouteTV predictions from data packets. The destination can determine the RouteTV for the route P by using formula(7). We assume that all nodes have a common Trust Table reference. If the received route is more trusted than the one currently in use, the destination sends a FLOW-
Fig. 4. The Flow Setup Process: (A) The Propagation Of FLOW-REQ Messages On Their Way To The Destination Node F; (B) The Intermediate Nodes Setup Their Flow States When They Receive The FLOW-SETUP Message.

SETUP message (e.g., all nodes in each route, node trust TV, route trust RouteTV, hop count) back to the source along the better chosen route. The intermediate nodes set up the flow state when they receive the FLOW-SETUP message. This flow setup process is shown in figure 4. In figure 4(a), the source node A sends a FLOW-REQ message for destination node F. Nodes B, C, D, and E will forward the FLOW-REQ messages and append information such as their own node ID and TV obtained from the evaluating (or upstream) node that the message received from. Therefore in our example, two FLOW-REQ messages arrive at node F. One contains route (A→B→C→E→F) with TV = (0.86, 0.91, 0.88, 0.89), and the other contains route (A→B→D→E→F) with TV = (0.86, 0.74, 0.92, 0.89). Node F can then compute the RouteTV for each route, the upper one is 0.87 and the later one is 0.885. It is easily to see that, route (A→B→C→E→F) is more trusted than route (A→B→D→E→F) since it has a larger route trust value (RouteTV 0.87) with the same hops.

(ii.) Route update

An RU packet contains the following fields: {SourceAddr, DestAddr, UpdateAddr, ChangeAddr, UpdateTrustValue}, where SourceAddr is the source node s, DestAddr is the destination node d, UpdateAddr is the update ID (i.e., the evaluating or monitored node k) and UpdateTrustValue in the RU is the update of TrustValue in the route. If the increase or decrease of trust in node v_k evaluated by node v_i is greater than or equal to the trust update threshold δ, i.e. \( \Delta TV_{jk} \geq \delta \) (node v_i and node v_j are in an existed forwarding route, and v_i is the next hop of v_j).

(iii.) Route Handoff

When the destination has detected that route trust of the used route is about to un-trusted, is reached, a FLOW-HANDOFF message is generated by the destination and propagated via flooding in the same way as the FLOWREQ message. After the source receives a FLOW-HANDOFF message, it determines the best route on which to handoff the flow based on the information contained in the FLOW-HANDOFF message (e.g., RouteTV, number of hops, etc.). The source then sends a FLOW-SETUP message along the new route. Note that this FLOW-SETUP message is the same as the one used in the FLOW-SETUP phase described previously in figure 4, except now it travels from the source to the destination. The handoff process is shown in figure 5 below.

Fig. 5. The Handoff Process: (A) The Original Route, With Nodes A and F Being The Source And The Destination; (B) The New Route After The Handoff Is Completed.

(iv.) Route error

When a link failure is detected (by a link layer feedback, for example), an RERR is sent back to the source s using that broken link. An unused route to a destination in a certain period is considered invalid and will be deleted from the source nodes’ routing cache. Specifically speaking, if the packet is retransmitted via some hop in the maximum number of times and no receipt confirmation is received, this node returns a Route Error message to the source s of the packet, identifying the link over which the packet could not be forwarded. Since at each hop, the host transmitting the packet for that hop can determine if that hop of the route is still working. If the data link level reports a transmission problem for which it cannot recover (for example, because the maximum number of retransmissions it is willing to attempt has been exceeded), this host sends a Route ERROR packet to the original sender s of the packet encountering the error. The Route ERROR packet contains the addresses of the hosts at both ends of the hop in error: the host that detected the error and the host to which it was attempting to transmit the packet on this hop. When a route error packet is received, the hop in error is removed from this host’s routing cache, and all routes which contain this hop must be truncated at that point. For sending such a retransmission or other packets to the same destination d, if source s in its own routing cache has another route to d (for example, from earlier route discovery, or from having overheard sufficient routing information from other packets), it can send the packet using the new route immediately. Otherwise, it may perform a new route discovery for this target.

c. Punishing Malicious Nodes

When a node exists in the black lists of all its neighbours, it will be excluded from the local network. Punishing malicious
nodes for a specific time is a solution for the problem of dynamic modification of a node’s behaviour. Every node in the black-list has a specific time (termed as the isolated time) in which the evaluated node \( v_i \) is regarded as a malicious node by the owner (evaluating node \( v_j \)) of the black-list. During the isolated time, node \( v_i \) is insulated from forwarding packets. After the time, node \( v_i \) will be removed from the black-list and its trust will be set to the black-list trust threshold. Node \( v_i \) will get a chance if node \( v_j \) has a packet to forward. If the packet is forwarded correctly by node \( v_i \), the trust will increase. If the packet is not forwarded correctly, node \( v_i \) will be put into the black-list again and be insulated for another term of the isolated time. The node’s trust prediction mechanism provides a MANET with the abilities against several attacks from malicious nodes, including on-off attack, conflict behaviour attack, modification attack, packet dropping (blackhole) attack, selective forwarding (grayhole) attack and so on [3],[15],[4]. However, a MANET is vulnerable to the Sybil attack and newcomer attack. If a malicious node can create several faked IDs, the trust prediction system might suffer from the Sybil attack [17],[8].

IV. DEVELOPMENT ENVIRONMENT

A. Experimental Setup

NS-2 simulator (Version 2.35) [6] is used to evaluate the performance of these reactive routing protocols in different conditions. The Distributed Coordination Function (DCF) of IEEE 802.11[20] for wireless LANs is adopted as the MAC layer protocol. We take an un-slotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) [16] to transmit data packets as well routing packets. Within a rectangular field of 1000 m x 1000 m, we dispersed 30 nodes randomly whose transmission radius of every node in one hop is fixed at 250 m. The node mobility uses the random waypoint model in which each packet starts its journey from a location to another at a randomly chosen speed. A maximum speed of 0 m/s implies that the MANET is a static network. Once the destination is reached, another destination is randomly chosen after a pause time. The fixed simulation parameters in NS-2 are listed in Table 5. Table 6 shows the simulation parameters in ETSR which are set as follows: The weights for control packets and data packets would be set according to two settings: (1) \( w_1 = 0.5 \) and \( w_2 = 0.5 \); (2) \( w_1 = 0.6 \) and \( w_2 = 0.4 \). In the setting (1), the forwarding ratio of control packets and data packets have equal importance. In setting (2), the forwarding ratio of control packets is considered more important than that of data packets. Accordingly, ETSR with setting (1) and (2) is represented as ETSR-1 and ETSR-2 respectively. And the black-list trust threshold is denoted by \( \eta \). It also makes a trade-off between trustworthiness and performance of the network, as the higher values of \( \eta \) ensure a more trustworthy network. It is possible for certain shorter routes to be discarded in the route discovery phase, even if there is no malicious node on such routes.

<table>
<thead>
<tr>
<th>Table 5 Fixed Simulation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Simulation Time</td>
</tr>
<tr>
<td>Number of Nodes</td>
</tr>
<tr>
<td>Map Size</td>
</tr>
<tr>
<td>Mobility Model</td>
</tr>
<tr>
<td>Traffic Type</td>
</tr>
<tr>
<td>Transmission Radius</td>
</tr>
<tr>
<td>Packet Size</td>
</tr>
<tr>
<td>Connection Rate</td>
</tr>
<tr>
<td>Pause Time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6 Varying Simulation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test No.</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

A. Assumptions / constraints considered in the experiment

i. Presence of malicious nodes

ii. Each node should be able to detect behaviour exhibited by malicious nodes serving to indicate Sybil attacks and hence we assume each node possesses an Intrusion Detection System (IDS). IDS detects attacks from other nodes in its radio range.

iii. Threat information from the IDSs will be communicated using standard data packets with no special control transmissions.

iv. Normal nodes are simulated to cooperate adequately on the network as cooperation is naturally required for a functional MANET to exist.

A. Performance parameters

We use four metrics to evaluate the performance of these routing protocols [11], in which the first two metrics are the most important for best effort route and transmit protocols.

i. Packet delivery ratio: It is the fraction of the data packets delivered to destination nodes to those sent by source nodes.

ii. Average end-to-end latency: It is the average time taken by the data packets from sources to destinations, including buffer delays during a route discovery, queuing delays at interface queues, retransmission delays at MAC layer and propagation time.

iii. Routing packet overhead: It is the ratio of the number of control packets (including route request-reply/update/error packets) to the number of data packets.

iv. Network throughput: Network throughput indicates the amount of digital data transmitted per unit time from source to destination.

To decrease the disturbance of random error, every experiment repeats 30 times and the average experiment results are computed. The above defined parameters are to be used for comparing the proposed work with the existing
protocols such as TSR (Trust-based Source Routing protocol) and AODV (Ad-hoc On-demand Distance Vector Routing Protocol). These protocols are to be compared with the existing work against the following scenarios:

i. Varying the maximum speed of the node, and
ii. Varying the number of malicious nodes

V. RESULT ANALYSIS
The results of the performance analysis of the proposed system (ETSR) conducted with the existing system (TSR) and Ad hoc On Demand Distance Vector (AODV) protocol under two test scenarios are discussed as follows.

A. Test 1: Varying node speeds
From the figure 6(a), we can see that the delivery ratio of the AODV drops remarkably as the maximum speed of the nodes increases, while TSR, ETSR1 and ETSR2 drops gently. It differ more at the higher speeds. This advancement can attribute to improved behaviour of detection due to the trust factor. The TSR, ETSR1 and ETSR2 can make use of the node’s trust feature, which evaluate the probability of successful delivery to the trusted nodes via trusted path. Whereas the AODV maintains one shorter route to the destination and is unable to improve the packet delivery ratio in case of route break or attacks. ETSR2 performs better than the TSR and ETSR1 because it evaluates the better trust values when compared to other protocols. Average end-to-end latency increases with increase in the maximum speed. At the maximum speed of 30m/s latency reaches its peak. At higher speeds the route makes a number of rediscoveries to find out the destination. TSR, ETSR1, and ETSR2 has a little lower average latency than AODV, because the trust based protocols (TSR, ETSR1 and ETSR2) avoids the malicious nodes, thus reducing the risk of adding delays for resending failed packets. This is shown in the figure 6(b). Routing packet overhead increases with increase in the maximum speed because the route links break down easily. TSR, ETSR1 and ETSR2 have higher routing packet overheads, this is because: i.) more number of Flow-REQ and Flow-SETUP packets are to be sent to qualified routes to meet the trust requirements, which is not considered in AODV. ii.) In addition, RouteUpdate packets increase the amount of control packets and hence increasing the routing packet overheads in TSR, ETSR1, and ETSR2. ETSR2 has smaller routing packet overheads than TSR and ETSR1 because it obtains a better trust values and maintains them in the routing cache when compared to the other two. This is clearly visible in the figure 6(c). Network throughput is directly proportional to the packet delivery ratio. At the speed of 10 m/s the throughput of AODV is 0.2 whereas the throughput of ETSR2 is 0.36, thus our approach improves the throughput to about 60%. Corresponding with figure 6(a) lower the packet delivery ratio lower the network throughput. This is shown in the figure 6(d).

B. Test 2: Varying number of malicious nodes
With no malicious nodes the packet loss rate is about 4%. The delivery ratio of AODV declines sharply, while TSR, ETSR1 and ETSR2 degrade gently as the number of malicious nodes increases. The delivery ratio of TSR, ETSR1 and ETSR2 are always higher than AODV, this is because using the trust, TSR, ETSR1 and ETSR2 allows no malicious nodes to forward packets. The delivery ratio of ETSR2 drops from 96% to 78% as the number of malicious nodes increases from 0 to 9. The reason for this drop is that with the proportion of malicious nodes increases, the probability of suspect or low trustworthy nodes existed on the routing route also increases, leading to descend the packet delivery ratio. From the sharp attenuation in AODV, we find that, malicious nodes make huge damage to the whole network, and more malicious nodes are, the more serious their damage is. This is clearly visible in figure 7(a). From figure 7(b), it is seen that the average end-to-end latency in TSR, ETSR1 and ETSR2 increases slowly as the number of malicious nodes increases. This average latency is mainly caused by queuing and retransmission delays. This reason is that, the TSR and ETSR add trust concept, along with the malicious nodes increase.
the routing route established by these methods may add hops, which results in the greater delay. However, the average latency in AODV ascends sharply and there is an obvious reduction in the average latency with TSR, ETSR1 and ETSR2 compared to AODV. There are two reasons: (1) in the process of route discovery and Path Selection, the network can avoid malicious nodes; (2) the availability of alternative routes eliminates delay caused by route rediscoveries in AODV, while multiple candidate mechanisms avoid route rediscoveries in TSR and ETSR, which contribute to effectively reduce the end-to-end latency. When the number of malicious nodes increases to 9 (30% of whole nodes), the routing packet overhead of AODV, TSR, ETSR and ETSR2 are approximately 0.47, 0.26, 0.25 and 0.24 respectively as shown in the figure 7(c). The routing packet overhead of ETSR is smaller than TSR and AODV. When the number of malicious nodes is smaller than 4, the routing packet overhead in TSR and ETSR is bigger than in AODV, the reason is that, the increased control packets in TSR and ETSR are primarily due to their route discovery mechanism that broadcasts more Flow-REQ and Flow-SETUP packets to look for trustworthy routes to destinations. However, when the number of malicious nodes is bigger than 5, the routing packet overhead in TSR and in ETSR is smaller than AODV, because of that the huge damage on routing path from malicious nodes. In AODV, due to the absence of the participation of the trust model, along with the increase number of malicious nodes, almost all of the routing route has the participation of malicious nodes which launch a constant 30% probability of modification attack, leading to the sharply increase in routing packet overhead. Figure 7(d) shows that our proposed approach can get an obvious throughput than TSR and AODV. Corresponding with figure 7(a) lower the delivery ratio lower is the network throughput. The experiment results in tests 1 and 2 show that ETSR performs better than TSR and AODV. With the help of ‘trust’, we use smaller increment of routing overhead to exchange with the bigger enhancement of the network security and performance, as ETSR gives higher delivery ratio, network throughput and detection ratio for malicious nodes.

Overall, ETSR2 shows a better performance than ETSR1. That proves our observation that control packets play a more important role than data packets in a MANET. For a real-world application, the factor values can be adjusted to satisfy the need of the trustworthiness and performance, depending on the requirement, characteristics, and environment of the application.

VI. CONCLUSION AND FUTURE WORK

An ad-hoc network is a set of limited range wireless nodes that function in a cooperative manner so as to increase the overall range of the network. Each node in the network pledges to help its neighbours by passing packets to and fro, in return of a similar assurance from them. All is well if all participating nodes uphold such an altruistic behaviour. However, nodes are subjected to a variety of attacks by other nodes. To address those demands, in this paper, we introduce a novel trust model, based on individual historical experience and logic rules prediction method. Taking the node’s prediction trust value as the input, a novel reactive trusted routing protocol extending from the standard Source Routing Mechanism, called Enhanced Trust-based Secured Source Routing protocol (ETSR) is proposed, which can improve the TSR to kick out the untrustworthy nodes such that a reliable passage delivery route is obtained and alleviate the attacks from malicious nodes. In this protocol, a source establishes optimal trustworthy routes in a single route discovery. This
protocol provides a flexible and feasible approach to choose a better route in all path candidates with trust constraint. Performance comparison of these routing protocols (AODV, TSR and ETSR) shows that ETSR is able to achieve a remarkable improvement in the packet delivery ratio, network throughput and defend some classical malicious attacks (e.g., fractions of modification, grayhole and blackhole attacks). For future work, we plan to incorporate other influencing attributes to the trust model. Moreover, other criterion can be used to determine the optimum route to set up the flow such as route Quality of Service (QoS), load and delay. We will explore these other factors to take a weighted average of the criteria into consideration when selecting a route in future works. In addition, the problem of dynamic behaviour modification will also be considered.

ACKNOWLEDGMENT

We gratefully acknowledge the guidance rendered by Mrs. G. Santhi, Assistant Professor, Department of Information Technology, Pondicherry Engineering College, Puducherry, India.

REFERENCES


