Detection of Malicious Attacks in Mobile Ad Hoc Networks (MANET)

G. Michael, R. Kavitha

Received: 01 April 2018 • Revised: 21 April 2018 • Accepted: 03 June 2018

Abstract: The highest security issue that is faced in current scenario in MANET is infrastructure centralization. The task routing and forwarding is achieved to the max only when the there is excellent cooperation among the participating nodes. By selfish act of few nodes, the process of forwarding is interrupted for reasons like saving the battery power. Malicious activities like Denial of Service (DoS) raise a security threat to MANET. Research and analysis of various security attacks is the focal point of this project. In this system DSR Protocol is extended with a feature of “node friendship”, which creates smooth cooperation in an adhoc environment. Based on the above work the concept of grey hole attack can be detected and eliminated.

Keywords: Ad Hoc Networks, Dynamic Source Routing (DSR), Route Request, Malicious Nodes.

INTRODUCTION

Mobile ad hoc networks (MANETs) operate in highly dynamic, infrastructure-less and potentially hostile environments, with limited bandwidth and energy resources. A typical MANET is shown in Fig 1.1 which comprises Laptops and Mobile devices. Thus, it is desirable to adaptively allocate these resources, so that network-level performance requirements are met—in spite of inherently unreliable wireless channels and ever-changing network topology.

Figure 1: Mobile Adhoc Network

AIM AND OBJECTIVES

A. Primary Objective

To apply trust based route selection to the Dynamic Source Routing (DSR) protocol, in order to fortify the protocol and improve route selection, which can increase throughput in situations where malicious nodes are present in the network.

1. Examine Existing ad hoc Routing Protocols

To attain knowledge and insight in the area of ad hoc routing existing protocols must be examined.

2. Investigate Security Solutions Applied to ad hoc Networks and ad hoc Routing Protocols

Existing security solutions that have been applied to ad hoc networks and routing protocols must be investigated, to identify useful methods and approaches.

G. Michael, Associate Professor, Department of Computer Science and Engineering, BIST, BIHER, Bharath Institute of Higher Education & Research, Selaiyur, Chennai. E-mail: michaelcse@gmail.com

R. Kavitha, Associate Professor, Department of Computer Science and Engineering, BIST, BIHER, Bharath Institute of Higher Education & Research, Selaiyur, Chennai
3. Research the Area of Trust

Research the area of trust management and trust in general to gain deeper knowledge of trust as a concept and to find formal methods for expressing trust.

B. Main Objectives

The main objectives are Analyze the DSR protocol, Design and implement components to incorporate trust based route selection to the existing DSR protocol. The Post objective needs to be fulfilled to determine to what extend the primary objective has been fulfilled and to identify possible areas of improvement. Simulations with the implemented extension have to be carried out. The results must be analyzed to determine the impact of applying the trust based routing strategies and to detect possible areas of improvements.

ANALYSIS of DSR

A. Sending of Packets

When a node has to send a package it queries its route cache for a route. If no route is found it broadcasts a ROUTE REQUEST. It is a possibility to perform a ring search as described in section 2B. Since ring search will limit the number of routes that are actually discovered and thereby increase the risk that all returned routes to a node include malicious nodes this might not be a good approach to use.

B. Selection of the “Best” Route

DSR selects routes from the cache based on the number of hops on the route, favoring short routes. This route selection strategy must instead be based on trust heuristics. An important observation is the fact that once DSR has a route to a destination it will use this route as long as it believes that it is working, meaning until it receives a ROUTE ERROR. This represents a serious problem in the case where only one route that contains a malicious node is available. It requires changes to DSR to poll for new routes if the routes available have to low a trust value and the consequences, such as increased ROUTE REQUESTs is somewhat unclear. Therefore introducing new mechanisms to discover new routes in the case where only one route containing a potential malicious node is available is considered out of scope of this assignment.

C. The use of DSR Acknowledgement Mechanism

As illustrated in Table 2-1 there is an option that allows the sender to set an acknowledgement bit in the DSR packet header in order to request an acknowledgement of the packet. If malicious nodes flip this bit it can prevent the receiver from returning acknowledgements or cause the receiver to send acknowledgements that was not requested. Since it is difficult to identify positive node behavior that could lead to an increase in trust without the use of acknowledgements there must be some mechanism for requiring acknowledgements. Therefore some extra fields are added to the packet header for this purpose.

D. Receiving Packets

This section describes and evaluates issues that can occur when a node receive a packet.

E. Forming Trust Relationships and Updating Trust

When a packet is received, the reversed route is stored in the cache. The route might contain unknown nodes which mean that an initial trust relationship with these nodes must be established. When a packet is received it also means that all nodes on the route actually forwarded the package correct. This knowledge is used to update trust values for the nodes on the route. There is a chance that the source of the route is malicious and does not forward packages. Therefore the trust value for the source of the route is not updated.

F. Route Reply’s

It is possible that malicious nodes return fictive non-useable routes in ROUTE REPLYS. One way to deal with this is to evaluate the received route based on the trust in the sender and not trust, route reply’s from nodes with low trust. This evaluation would however apply to all returned routes and cause extra computations and it would not solve the problem where a malicious node forged the reply so it seemed to be initiated by some other node. To avoid the risk that nodes alter routes, the entire route could be signed and the signature included in the payload as well as in the packet header. This would of course increase the size of the payload and it would require that cryptographic methods were available. Another method
for verifying the route is to let all nodes create a hash value based on the address of the node that forwarded the packet and thereby successive confirming each step of the route.

It is optional to reply to route replies from the route cache. By replying with a route from the route cache the number of forwarded route replies are limited which decreases the overhead on the network. On the other hand it might prevent new routes from being discovered. If only one route that includes a malicious node is contained in the cache this route is returned. This might represent a problem because it can cause bad routes to be distributed. Another issue is that the cache represents some a learned part of the topology. Some of the information stored in the cache might not be valid because nodes can have moved out of range, turned their power off, etc. This problem is known as cache staleness.

G. Forwarding of Packets

Issues of forwarding packets are mostly related to behavior of malicious nodes.

H. Drop of Forwarding Packet

All forwarding packets can be dropped. There is nothing that can be done to prevent that a nodes drops a packet. The Pathwardhan – Watchdog technique and the CONFIDANT protocol that was described in section 2.3.4 and 2.3.1, uses a monitor technique to detect when a node is not forwarded packages. These techniques do however not adequately solve the problems of framing and collision among malicious nodes and therefore it is decided to use another approach where nodes only base decision on their own experiences and not on events that other nodes inform them of.

I. Tampering with ROUTE REQUESTs

When forwarding a ROUTE REQUEST message a malicious node can choose not to add its own address to the address list, which will probably cause the route to be useless. There is no way for the initiator of the ROUTE REQUEST to directly detect that the route has passed through a node that has not added itself. A solution to verify that the address list is not defective has been proposed in the section ROUTE REPLYs were the reception of ROUTE REPLYs was discussed.

J. Tampering with the Address List

When forwarding a packet, a malicious node can choose to remove prior encountered nodes on the route from the address list or insert other nodes. This would most likely result in a useless route. This would especially be critical if the packet was a ROUTE REQUEST, since it eventually would result in a useless route. In order to avoid that nodes remove other nodes from the address list it would require each node to sign its own address in the list.

K. Snooping of Source Routes

DSR has the possibility of snooping routes, which means that when a node forwards or overhear a packet it can snoop the route and cache it for later use. If a snooped route contains unknown nodes initial trust relationships must be established.

Figure 2: If Node C Snoops the Route from D to A, IT Might Never Discover the Two Routes Going Through E and F

The use of route snooping will decrease the amount of ROUTE REQUEST packages that are send, but in some situations it might limit the number of routes that are discovered to a destination because nodes that has a route to a destination will not issue new ROUTE REQUESTs to discover new routes to that destination. Fig 3-1 illustrates how node C might not discover some routes. However it is likely that the route can be snooped from some other route. However it is chosen to allow route snooping.
EXISTING SYSTEM

The use of the DSR acknowledgement mechanism might not be suitable, since it is easy for malicious nodes to flip the acknowledgement bit and thereby tangle up the protocol.

All the protocols schemes presented above clearly have the overheads associated with the secure routing at all times. Our protocols will converge to the DSR protocol if all the nodes in the ad hoc network are friends. The routing overheads are significant in the presence of stranger nodes and acquaintance nodes.

A. The Assumptions Made about Malicious Nodes

1. Malicious nodes will not forward any packages for other nodes.
2. Malicious nodes will not return acknowledgements.
3. The malicious behavior specified above, means that malicious nodes does not forward route replies, but they do however Reply to route replies.

Good nodes can also drop packages unintended due to queue overruns, low battery etc.,

B. The Proposed System: Extended DSR Protocol

Identification of relationships between neighbours in an ad hoc network in an ad hoc network, the relationship of a node i to its neighbour node j can be any of the following types

i. Node i is a stranger to neighbour node j: Node i has never sent / received messages to/from node j. Their trust levels between each other will be very low. Any new node entering an ad hoc network will be a stranger to all its neighbours. There are high chances of malicious behaviour from stranger nodes.

ii. Node i is an acquaintance to neighbour node j: Node i has sent / received few messages from node j. Their mutual trust levels are neither too low nor too high to be reliable. The chances of malicious behaviour will have to be observed.

iii. Node i is a friend to neighbour node j: node i has sent / received plenty of messages to/from node j. The trust levels between them are reasonably high. Probability of misbehaving nodes may be very less. The above relationships are represented as a Friendship table for each node in an ad hoc network. Consider the node 7 in Fig 3-2. The friendship table of node 7 is represented as shown in Table 3-1. A trust estimator is used in each node to evaluate the trust level of its neighboring nodes. The trust level is a function of various parameters like length of the association, ratio of the number of packets forwarded successfully by the neighbour to the total number of packets sent to that neighbour, ratio of number of packets received intact from the neighbour to the total number of received packets from that node and average time taken to respond to a route request.

<table>
<thead>
<tr>
<th>Neighbours</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>S</td>
</tr>
</tbody>
</table>
The threshold trust level for a stranger node to become an acquaintance to its neighbour is represented by \( T^{\text{acq}} \) and the threshold trust level for an acquaintance node to become a friend of its neighbour is denoted by \( T^{\text{fri}} \). The relationships are represented as

\[
\begin{align*}
R(n_i \rightarrow n_j) &= F \text{ when } T \geq T^{\text{fri}} \\
R(n_i \rightarrow n_j) &= A \text{ when } T^{\text{acq}} \leq T < T^{\text{fri}} \\
R(n_i \rightarrow n_j) &= S \text{ when } 0 < T < T^{\text{acq}}
\end{align*}
\]

Also, the relationship between nodes is asymmetric. \( R(n_j \rightarrow n_i) \) is a relationship evaluated by node \( n_i \) based on trust levels calculated for its neighbour \( n_j \). \( R(n_j \rightarrow n_i) \) is the relationship from the friendship table of node \( j \). This is evaluated based on the trust levels assigned for its neighbour \( n_i \). This relationship is based on the combination of two algorithms called as discounting combination and consensus combination.

**ROUTING MECHANISM**

When any node wishes to send messages to a distant node, it sends the ROUTE REQUEST to all the neighboring nodes. The ROUTE REPLY obtained from its neighbour are sorted by time or number of packets forwarded to acquaintance and friend.

\[
K_{\text{af}} = \frac{\text{No. of packets forwarded to an acquaintance}}{\text{No. of packets forwarded to a stranger}}
\]

This is represented by a set \( S_1 \) which is a set of all discrete values of \( K_{\text{af}} \) assigned over a span of time \( (t_1 \text{ to } t_N) \)

\[S_1 = \{k_{\text{af}1}, k_{\text{af}2}, k_{\text{af}3}, \ldots, k_{\text{af}n}\} \quad 0 \leq k_{\text{af}} < 1\]

\( k_a \)- ratio defined at the \( i^{\text{th}} \) time interval.

\( k_a \)- threshold k ratio for an acquaintance.

The \( k_{\text{af}} \) value initially starts with \( k_{\text{af}1}(=k_a) \) for a specific interval of time and progresses with other discrete values. When \( k_{\text{af}} \) reaches the value \( k_{\text{af}} = 1 \), the acquaintance node can be upgraded to a friend node if the threshold trust level is satisfied.

<table>
<thead>
<tr>
<th>Next hop neighbour in the best path ( P_1 )</th>
<th>Next hop neighbour in the best path ( P_2 )</th>
<th>Action Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>( F ) is chosen in ( P_1 )</td>
</tr>
<tr>
<td>F</td>
<td>A</td>
<td>( F ) is chosen in ( P_1 )</td>
</tr>
<tr>
<td>A</td>
<td>F</td>
<td>( A ) or ( F ) based on ( k_a )</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>( A ) is chosen in ( P_1 ) after invoking the trust estimator</td>
</tr>
<tr>
<td>F</td>
<td>S</td>
<td>( S ) is chosen in ( P_1 )</td>
</tr>
<tr>
<td>S</td>
<td>F</td>
<td>( S ) or ( F ) based on ( k_a )</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>( S ) is chosen in ( P_1 ) after invoking the trust estimator</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>( A ) is chosen in ( P_1 ) after invoking the trust estimator</td>
</tr>
</tbody>
</table>

**K_{\text{af}}** is the ratio between packets forwarded to stranger and friend

\[
K_{\text{af}} = \frac{\text{No. of packets forwarded to an acquaintance}}{\text{No. of packets forwarded to a stranger}}
\]

This is represented by another set \( S_2 \)

\[S_2 = \{k_{\text{sf}1}, k_{\text{sf}2}, k_{\text{sf}3}, \ldots, k_{\text{sf}n}\} \quad 0 \leq k_{\text{sf}} < k_a\]

\( K_{\text{sf}} \) is ration at the \( i^{\text{th}} \) interval of time.

The \( K_{\text{sf}} \) value starts with \( K_{\text{sf}1}(=0) \) for a stranger and progresses with assignment of discrete ratios. When \( k_{\text{sf}n} = k_a \), the node is an eligible candidate to be promoted to an acquaintance based on the trust levels.

Similarly, the ratio \( K_{\text{sa}} \) denotes the number of packets forwarded to a stranger and an acquaintance.

\[
K_{\text{sa}} = \frac{\text{No. of packets forwarded to an acquaintance}}{\text{No. of packets forwarded to a stranger}}
\]

This is represented by another set \( S_3 \).

\[S_3 = \{k_{\text{sa}1}, k_{\text{sa}2}, k_{\text{sa}3}, \ldots, k_{\text{sa}m}\} \quad 0 < k_{\text{sa}} < 1\]
k_{val} = ratio at the \( i^{th} \) interval of time.

The source selects the shortest and the next shortest path. Whenever a neighboring node is a friend, the message transfer is done immediately. This eliminates the overhead of invoking the trust estimator between friends. If it is an acquaintance or stranger, transfer is done based on the corresponding ratios \( kaf, ksa \) and \( ksf \). The trust estimator is invoked. This protocol will converge to the DSR protocol if all the nodes in the ad hoc network are friends. Further the overheads due to the calculations of trust relationship are minimal compared to the CONFIDANT protocol. It will be slightly more than the normal DSR due to the invocation of the trust estimator whenever a data transfer is to be done through strangers or acquaintances. The messages are transmitted to the distant node in the above mentioned way by selecting the route at each intermediate node. The \( k \) parameters at discrete time intervals are design parameters. Simulation is carried out with suitable values for all the parameters and the threshold thrust levels so as to obtain optimum performance. There is a tradeoff between offering good security in ad hoc networks and overall throughput of the network. Hence, choosing an optimal value is crucial for the good functioning of the network.

**SIMULATION**

*A Simulation tool: OPNET [8]*

**Scenario: Grey hole at Mobile nodes 5 and 3 with Mobility**

1. This Scenario shows a 9 node DSR network.
2. All nodes in the network are configured to run DSR.
3. The Mobile Node 5 is acting as a Grey Hole Node.
4. The Mobility and Trajectory is defined for the Mobile Nodes 0, 1, 2, 4, 5, 6 and 8.
5. All the mobile Nodes have attribute \( t \) value and \( t \) state initially values are \( t \) value = 0 \( t \) state = " Stranger"

<table>
<thead>
<tr>
<th>Mobile_node</th>
<th>IP Address</th>
<th>Trust Value</th>
<th>Trust State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile_node_0</td>
<td>192.0.0.1</td>
<td>5</td>
<td>Friend</td>
</tr>
<tr>
<td>Mobile_node_1</td>
<td>192.0.0.2</td>
<td>5</td>
<td>Friend</td>
</tr>
<tr>
<td>Mobile_node_2</td>
<td>192.0.0.3</td>
<td>0</td>
<td>Stranger</td>
</tr>
<tr>
<td>Mobile_node_3</td>
<td>192.0.0.4</td>
<td>5</td>
<td>Friend</td>
</tr>
<tr>
<td>Mobile_node_4</td>
<td>192.0.0.5</td>
<td>5</td>
<td>Friend</td>
</tr>
<tr>
<td>Mobile_node_5</td>
<td>192.0.0.6</td>
<td>0</td>
<td>Stranger</td>
</tr>
<tr>
<td>Mobile_node_6</td>
<td>192.0.0.7</td>
<td>2</td>
<td>Acquaintance</td>
</tr>
<tr>
<td>Mobile_node_7</td>
<td>192.0.0.8</td>
<td>5</td>
<td>Friend</td>
</tr>
<tr>
<td>Mobile_node_8</td>
<td>192.0.0.9</td>
<td>5</td>
<td>Friend</td>
</tr>
</tbody>
</table>
CONCLUSION AND FUTURE WORK

An ad hoc routing protocol, DSR is extended with some trust values and functions and identified the malicious nodes (Grey hole) there by implementing Trust based routing. Hence the efficiency of the DSR Protocol is improved, but it will cause little bit of overheads as well. In the proposed system we have identified the malicious nodes that can be eliminated by using the Trust based routing through Extended DSR. So in the future we planned to introduce more number of malicious nodes and eliminate the same in routing and thereby measuring its performance.

REFERENCES


