**TRUSTWORTHY MANET ROUTING ESTAODV IMPLEMENTATION USING DEEP REINFORCEMENT LEARNING**

Project ID: 18-024



**Final Report**

Jude Niroshan

Anushka Kodithuwakku

Sajith Chamara Gunawardene Liyanage

Sadunika Kularatne

Bachelor of Science Special (Honors) in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2018

# **DECLARATION**

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

**Authors:**

|  |  |  |
| --- | --- | --- |
| Name | Registration No | Signature |
| M.A.J. Niroshan | IT14085918 |  |
| K.A.H. Kodithuwakku | IT15564078 |  |
| Liyanage S.C.G. | IT14098888 |  |
| M.A.S.H. Kularatne | IT15143914 |  |

The above candidates are carrying out research for the undergraduate dissertation under my supervision.

**Supervisor:**

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Mr. Prabath Lakmal Rupasinghe |  |  |

# **Abstract**

A collection of nodes which have the ability to move randomly within a wireless network is called a mobile ad hoc network. Mobile Ad Hoc Network plays a major role in wireless communication technology. Data transferring within the network has two considerable facts, reliability and security. Ensuring security in a mobile ad hoc network is a major concern due to the unpredictable motions and behaviors of network nodes.

In a wireless mobile network, it is possible for a large number of data packets to transmit among nodes within a small period of time. Therefore, it is possible that some nodes might not behave as expected. It can eventually cause a considerable amount of data packet drops. It shows that the existing security mechanisms have failed to distinguish between trustworthy and malicious nodes. Generally, the nodes select the shortest path; but sometimes it may not be the reliable route to transfer data. Proposing a method to evaluate the trustworthiness of each node, will address the issue up to a reasonable extent.

Network simulators can model the behaviors of an actual network. There are two types of simulators called CLI (Command-Line Interface) driven and GUI (Graphical User Interface) driven simulators. Analyzing the network security and performance metrics after executing the proposed trust-based schema would help us to understand the importance of a trust-based schema for Mobile Ad-hoc Networks (MANET). NS-3 is a CLI based simulator which can be used for the simulation purposes.

Keywords— MANET, global trust, trust framework, spiral model, RL component

# **ACKNOWLEDGEMENT**

The authors would like to express special thanks of gratitude to their project supervisor, Mr. Prabath Lakmal Rupasinghe for his patient guidance and useful critique of their research work. And the authors are grateful for the corporation given by Ms. Chamira Nawarathna to make the research project a success. Their willingness to generously give their time, the moral support and their continuous guidance by providing important feedback have been very much helpful and valuable for the authors to successfully complete their research work. Authors would also like to thank their lecturer-in-charge for the course module, Mr. Jayantha Amararachchi for his constant guidance and assistance to carry out this research project. Also, the authors would like to thank their parents and friends who supported and encouraged them throughout the time period of this research project.

# **TABLE OF CONTENTS**

[**DECLARATION** 2](#_Toc526965894)

[**Abstract** 3](#_Toc526965895)

[**ACKNOWLEDGEMENT** 4](#_Toc526965896)

[**TABLE OF CONTENTS** 5](#_Toc526965897)

[**LIST OF FIGURES** 6](#_Toc526965898)

[**LIST OF TABLES** 6](#_Toc526965899)

[**LIST OF ABBREVIATIONS** 7](#_Toc526965900)

[**1** **INTRODUCTION** 7](#_Toc526965901)

[**1.1** **Background** 8](#_Toc526965902)

[**1.2** **Literature Review** 8](#_Toc526965903)

[**1.2.1** **Black hole attack in AODV routing protocol [1]** 9](#_Toc526965904)

[**1.2.2** **Authentication using trust to detect misbehaving nodes in mobile ad Hoc networks using Q-Learning [1]** 9](#_Toc526965905)

[**1.2.3** **Information theoretic framework of trust modeling and evaluation for ad hoc networks [2]** 10](#_Toc526965906)

[**1.2.4** **Different ways to achieve trust in MANET [3]** 11](#_Toc526965907)

[**1.2.5** **Trust management in mobile ad hoc networks using a scalable maturity-based model [5]** 12](#_Toc526965908)

[**1.2.6** **Secure routing with AODV protocol for mobile ad hoc networks [6]** 13](#_Toc526965909)

[**1.2.7** **QoS assertion in manet routing based on trusted AODV (ST-AODV) [7]** 13](#_Toc526965910)

[**1.2.8** **EBoX: Evidence of behavior information exchange mechanism against selfish attacks [8]** 14](#_Toc526965911)

[**1.2.9** **QoS of MANET through trust based AODV routing protocol by exclusion of black hole attack [9]** 14](#_Toc526965912)

[**1.3** **Research Gap and Research Problems** 15](#_Toc526965913)

[**1.4** **Research Objectives** 16](#_Toc526965914)

[**2** **METHODOLOGY** 17](#_Toc526965915)

[**2.1** **Direct Trust Calculation** 17](#_Toc526965916)

[**2.2** **Indirect Trust Calculation** 19](#_Toc526965917)

[**2.3** **Identifying Trust Levels** 22](#_Toc526965918)

[**2.4** **Spiral Model** 24](#_Toc526965919)

[**2.4.1** **Collaborative malicious node discovery process** 25](#_Toc526965920)

[**2.4.2** **Penalty phase** 26](#_Toc526965921)

[**2.4.3** **After transmission phase** 27](#_Toc526965922)

[**2.5. Deep Reinforcement Learning Model** 29](#_Toc526965923)

[**2.6 Trust based routing protocols framework** 30](#_Toc526965924)

[**2.7 Performance Metrics Collection Component** 31](#_Toc526965925)

[**2.8.**  **Technical Objectives (S/W and H/W Requirements)** 32](#_Toc526965926)

[**2.8.1. Technologies** 32](#_Toc526965927)

[**2.8.2. Software Requirements** 32](#_Toc526965928)

[**2.8.3. Hardware Requirements (Minimum)** 33](#_Toc526965929)

[**3. Results and Discussions** 34](#_Toc526965930)

[**3.1. Vary the number of malicious nodes** 36](#_Toc526965931)

[**3.1.1. Packet Delivery Ratio** 36](#_Toc526965932)

[**3.1.2. End to End Delay** 37](#_Toc526965933)

[**3.1.3. Packet Drop Ratio** 37](#_Toc526965934)

[**3.2. Malicious Node Detection Accuracy** 38](#_Toc526965935)

[**4.** **SUMMARY OF CONTRIBUTION** 40](#_Toc526965936)

[**5.** **CONCLUSION** 44](#_Toc526965937)

[**6.** **REFERENCES** 45](#_Toc526965938)

# **LIST OF FIGURES**

[Figure 1.1: Sample Network Diagram with 3 Network Nodes 10](file:///C:\Users\SajithChamara\Desktop\18-024_FinalThesisSoftBound.docx#_Toc526967137)

[Figure 1.2: Node A Receives Recommendations about Node D [5] 12](file:///C:\Users\SajithChamara\Desktop\18-024_FinalThesisSoftBound.docx#_Toc526967137)

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
|  |  | Page |
| Table 3.1 | Trust Table | 12 |
| Table 3.2 | Threshold Table | 16 |
| Table 3.3 | Recommendation Table | 17 |
| Table 3.4 | Backup Table | 17 |

# **LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| Abbreviation | Description |
| ABED | Ant Based Evidence Distribution |
| AODV | Ad hoc On Demand Distance Vector |
| CP | Control Packets |
| DP | Data Packets |
| DT | Direct Trust |
| GRE | Generalized Reputation Evidence |
| IDM | Intrusion Detection Model |
| IRM | Intrusion Response Model |
| MANET | Mobile Ad hoc Network |
| NFC | Near Field Communication |
| QoS | Quality of Service |
| REP | Recommendation Exchange Protocol |
| RERR | Route Error |
| RL | Reinforcement Learning |
| RREQ | Route Request |
| TL | Trust Level |
| TRR | Trust Recommendation Request |

# **INTRODUCTION**

## **Background**

Wireless communication is a communication mode which does not use physical wires to connect between two or more devices to transfer data. It uses electromagnetic waves to transfer signals. Depending on the wave frequencies, network coverage area will be changed. It can cause network connectivity issues for some regions. Generally, there are more advantages of using wireless networks. Cost is low since it does not require any physical infrastructure to maintain. Frequently, flexibility and accessibility of a wireless network are high, regardless of the location. Some of the popular wireless technologies are WiFi, Bluetooth, NFC (Near-field communication) and satellite services. Mobile ad hoc network is a dynamic network which uses wireless technologies. MANET is widely used in applications such as home intelligence devices, military and Sensor networks [3], [11]. Routing protocols specify how routers should communicate with each other in the network. In a mobile ad hoc network, the ad hoc routing protocol is used for this purpose.

Due to the mobility feature of network nodes in the mobile ad hoc networks, security issues could arise at any time such as the packets being dropped due to some unpredictable conditions. Therefore, the regular transmission process of the network can be interrupted. Existing cryptographic techniques like public/private key encryption and other security mechanisms such as packet filters, firewalls cannot always identify the trustworthy nodes to communicate. In public/private key encryption, anyone can encrypt a message using the public key of the receiver. Packet filtering or static filtering is a firewall technique which can monitor, based on incoming and outgoing packets. As diverse to all the above-mentioned methods, defining a trust-based schema on top of ad hoc networks to detect each one hop (directly connected) neighbor nodes will solve this issue up to a considerable degree.

## **Literature Review**

Even though there are some existing trust-based schemas to detect malicious nodes and isolate from the network, they were not implemented to define the exact behaviors of malicious nodes.

### **Black hole attack in AODV routing protocol [1]**

One of the most popular attacks in AODV (Ad hoc On-Demand Distance Vector) routing, is the ‘Blackhole attack’. When the source needs to send packets to an unknown destination which is not included in sender’s routing table, it will send RREQ (route request) packets to neighbor nodes to identify the destination node. In that case, a black hole node can send the reply packet (RPLY) before all the other nodes, it will be chosen as the route to transfer data. Black hole node receives the information in the packets and drops them without passing towards the destination. Since the sender does not suspect neighbor nodes in the existing network, it will continue to communicate with the harmful black hole node. Eventually, that leads to making the packet dropping ratio a higher value.

Apart from RREQ and RPLY, there are a different number of other control packets are used in AODV protocol. RERR (route error) message is sent by neighbor nodes whenever a link failure happens or destination is unreachable. Basically, HELLO packets are broadcasted to identify the available neighbor nodes in the network. After a time period, we can count the number of forwarded and received data packets. Using that information, we can calculate the trust between directly connected nodes.

### **Authentication using trust to detect misbehaving nodes in mobile ad Hoc networks using Q-Learning [1]**

Authentication which is the key factor to be considered in MANET can be categorized into two sections called pre-authentication and post-authentication. As the name denotes pre-authentication is initial network deployment and post-authentication is a mechanism to detect nodes in the network over a period of time. According to S.Sivagurunathan, K.Prathapchandran, and A.Thirumavalavan, trust can be defined as “*the reliability, timeliness, and integrity of message delivery to a node’s intended next hop*” [1].

Nodes in the ad-hoc network will eventually be categorized into three sections such as trustworthy, partially trusted and untrusted; based entirely on their direct trust. Untrusted category contains types of nodes such as selfish [10], pure malicious and collaborative malicious nodes. So, it is not advisable to come to conclusions based only on their direct trust value. There could also exist indirect aspects throughout the network which might affect the trust between nodes. In that case, apart from the direct trust, an indirect trust value which would consider such indirect factors should be calculated. Subsequently, a global trust value can be defined based on the average value of both direct and indirect trust values and that global trust can be used for the rewarding system within the network.

### **Information theoretic framework of trust modeling and evaluation for ad hoc networks [2]**

It is preferred to consider the recommendation values from other nodes to fulfill the requirement of calculating indirect trust. Yan Lindsay Sun, Wei Yu, Zhu Han, and K.J. Ray Liu have proposed an information theoretic framework as a solution. According to them, trust is a “*measure of uncertainty with its value represented by entropy”* [2].

This is a better approach than the 1.2.2 solution to detect misbehaviors of nodes because it defines a combination of two trust models named ‘entropy-based model’ and ‘probability-based model’. Under the entropy-based model, they have introduced an equation to calculate TABC which is the same as the indirect trust between node A and C.

Figure SEQ Figure \\* ARABIC 1.1: Sample network diagram with 3 network nodes



Figure 1.1: Sample Network Diagram with 3 Network Nodes

RAB is the recommendation value from node A to B and TBC is the trust value from node B to C. Probability-based model will calculate the multipath trust propagation and concatenation using probability equations. Probability values for the trust relationship can be converted into trust values using entropy-based equations. In order to calculate indirect trust, it is required to request recommendations from other nodes. A new control packet has been introduced as TRR (Trust Recommendation Request) to get the trust value of a particular node by requesting from the other neighbor nodes.

TABC = RABTBC

According to the Figure 1.1, if node A wants to know the indirect trust value of node C, node A can send a TRR message to node B by requesting for the trust value of node C.  That trust value is only available in node B’s trust table. Finally, TABC can be evaluated as in the above equation. Based on that trust value, their attempt is to detect malicious nodes.

One drawback of this solution is that malicious nodes can collaboratively provide wrong recommendations for other nodes. Therefore, a mechanism is required to detect collaborative malicious nodes. By analyzing this history of network node interactions, we provided a solution to categorize nodes into levels based on the global trust which can be utilized to identify the malicious nodes. Network nodes can be trustworthy, partially trustworthy, selfish, pure malicious or collaborative malicious nodes.

### **Different ways to achieve trust in MANET [3]**

It is important to study the existing trust calculation schemes. Each scheme has different unique features, merits, and findings. There are five main trust schemes in MANETs.

* Protocol-based scheme - Basically, security protocols have been implemented in this scheme. ABED (Ant-Based Evidence Distribution) scheme utilizes a swarm intelligence paradigm [4] to model the protocol-based schema. Communication among network nodes occurs through agents similar to ants [3] in ABED. Ants collected information which is called as pheromones [3]. Based on pheromones, ants find an optimal path to measure trust evidence. Generalized Reputation Evidence (GRE) [3], is another instance of the protocol-based scheme.
* System level-based scheme - Under system level-based, it will give rewards to trustworthy nodes and give a penalty to malicious or selfish nodes. Because of this purpose, they have defined some trust models. Watchdog trust model can detect selfish nodes and Collaborative Reputation trust scheme will distinguish between selfish nodes and malicious nodes.
* Cluster based scheme
* Maturity based scheme
* PKI (Public Key Infrastructure) based scheme

### **Trust management in mobile ad hoc networks using a scalable maturity-based model [5]**

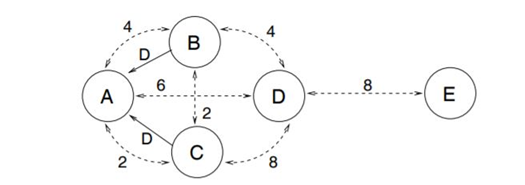
Recommendation exchange protocol (REP) is used to send and receive recommendations via the network. Basically, a network node will broadcast messages to its directly connected nodes (neighbor nodes) requesting for the recommendation value regarding a particular neighbor node.

Figure 1.2: Node A Receives Recommendations about Node D [5]

Figure 1.2 indicates that node B and node C will send REP to node A along with the recommendation value for node D. Each of these dotted arrows in the diagram are represented with a number which denotes the time period; how long they have known each other. When accepting the given recommendation value, node A will give a higher priority for node C than node B since node C has known node D for a longer period of time.

### **Secure routing with AODV protocol for mobile ad hoc networks [6]**

T. Farid and A. Prahladachar have basically defined two types of security attacks and two types of models as the proposed solution. Compromised network nodes and selfish network nodes could make ‘internal attacks. When a network node does not send or forward data packets and become inactive when other nodes need them and become active only for its own benefits, it can be named as a ‘selfish node’. There is also another type of attack called ‘external attack’ which occurs due to invalid cryptographic information. Intrusion Detection Model (IDM) and Intrusion Response Model (IRM) comes into the front as the solution. IDM uses neighbor node information to detect misbehaving nodes. If misbehavior count is greater than the threshold value, it will broadcast that misbehaving node to other nodes. Under IRM, if two or more network nodes report about the same node, Purge packet [6] is transmitted to isolate the malicious node from the network.

### **QoS assertion in manet routing based on trusted AODV (ST-AODV) [7]**

In order to increase Quality of Service (QoS) [12] in ad hoc networks, Sridhar Subramanian and Baskaran Ramachandran have proposed a trusted AODV called “ST-AODV”. The trust level (TL) value is calculated as below.

**TL = T(RREQ)\*Qr + T(RREP)\*Qp + T(DATA)\*Qd**

Their Qr, Qp, and Qd are the intermediate values to calculate the request rate, reply rate and data transmission rate of network nodes respectively. And time factor to evaluate the route request, route reply and data sent are measured via T(RREQ), T(RREP) and T(DATA) accordingly. For a given network node, if TL is less than or equals a threshold value, then it is considered as an untrustworthy node who might drop packets. Otherwise, it is a trustworthy node who should be allowed to stay in the network for a better-secured communication.

### **EBoX: Evidence of behavior information exchange mechanism against selfish attacks [8]**

Evidence of Behavior Information Exchange (EBox) mechanism could detect selfish nodes and misbehaving nodes in the ad hoc network and eliminate such nodes from the network. It uses a reputation-based schema to award penalty points for selfish network nodes and credit points for trustworthy network nodes. Santhosh J and Malini V K proposed this mechanism to estimate trust values by comparing their predefined threshold values.

### **QoS of MANET through trust based AODV routing protocol by exclusion of black hole attack [9]**

Radha Krishna Bar, Jyotsna Kumar Mandal, and Moirangthem Marjit Singh have proposed to evaluate trust values for nodes in the ad hoc networks based upon two criteria; their ability to forward data packets and forward RREQ for a given network node. Finally, the trust value is calculated as a multiplication of the forwarded data packets ratio and forwarded RREQ packets ratio. Then trust value is recorded in the routing table to make routing decisions.

## **Research Gap and Research Problems**

Nodes in MANET can move randomly without any centralized structure or any time pattern. Due to this self-configuration and self-optimization characteristics, such networks can be called self-organized networks [1]. It is difficult to provide security for such dynamic environments than traditional networks. Ad hoc networks like MANET are vulnerable to various attacks due to this dynamic and distributed behavior of nodes. This can lead to many IoT device failures in resource-constrained environments. Therefore, there should be mechanisms which allow a node to measure the reliability and security of other nodes. Then trustworthy nodes can avoid dealing with malicious nodes. As a result, it can improve both network performance and security aspects.

As revealed in 1.2.2, only the direct trust is calculated to evaluate the trustworthiness of nodes that will cause problems in capturing indirect behaviors of harmful network nodes. There is no way to prove complete trustworthiness only depending on direct interactions among each node in the network. There can be instances of achieving high accuracy for trust values by getting recommendations from other network nodes. At the same time, one cannot come to a better decision only depending on indirect trust value. This highlights the requirement of calculating the average value of direct trust value and indirect trust value when arriving at a more informed conclusion on the trustworthiness of nodes. On the other hand, the definition of trust among the network nodes is similar to trust among human beings. Direct trust is the trust which is built with the experience among each other. When someone suspects trust, they take recommendations from others which is the indirect trust. Therefore, measuring both direct trust and indirect trust is a vital factor.

According to 1.2.3, they do not consider the collaborative behaviors of malicious nodes. Sometimes a group of malicious nodes provide wrong recommendations to make a node in their team as more trustworthy. Eventually, it also contributes to a considerable amount of packet drops. Then there should be categories of malicious nodes such as pure malicious and collaborative malicious. Pure malicious nodes will misbehave individually, while collaborative malicious nodes misbehave as a team in the network. Therefore, it is important to distinguish the type of malicious nodes.

## **Research Objectives**

**Main Objectives**

* Categorize network nodes into defined trust levels according to their trustworthiness.
* Dynamically detect malicious nodes and isolate them from the network.
* Analyze performance metrics when detecting the secured and reliable routes using proposed trust-based schema.
* Analyze historical trust value fluctuations and calculate a Q-value based on dynamic reward points for each of the node within the trust table.

**Specific Objectives**

* Calculate direct trust and indirect trust values to measure the overall trustworthiness of network nodes.
* Deduct penalty points from trust value for possible malicious activities of directly connected nodes.
* Distinguish pure malicious nodes and collaborative malicious nodes in the ad hoc network.
* Implement Trust Recommendation Request (TRR) algorithm to exchange trust recommendations among directly connected network nodes.
* Predict the secured routes in MANET using Q-Learning mechanism.
* Implement a generic trust-based routing protocols framework to facilitate trust routing protocol implementations.

# **METHODOLOGY**

## **Direct Trust Calculation**

Over the period of time, nodes are required to send packets to a particular destination. All the nodes broadcast the HELLO packets to discover the desired route. Only one hop neighbors respond to the hello packets because they are in the same communication range. So that each node ensures it’s one hop neighboring nodes ultimately. From that, each node can conclude how many nodes are staying as one-hop neighbors. In AODV the following control packets are used. In the route discovery phase, route request (RREQ), route reply (RPLY) packets are used. Route Error (RERR) and HELLO packets are used in route maintenance phase. While evaluating trust, these packets are also considered because they provide a significant contribution towards the routing operations. Though misbehaving nodes can also utilize such packets, the probability of utilization of such packets is relatively low compared with the well-behaving nodes.

NDF denotes the number of data packets actually forwarded, NDR denotes the number of packets actually received over a period of time with n number of interactions. Likewise, every node could calculate the trust value of all its one-hop neighbors and update its Trust table. Each node can monitor its neighboring nodes’ forwarding behavior by using passive acknowledgment. After calculating the Control Packet (CP) and the Data Packet (DP), Direct Trust (DT) is calculated.

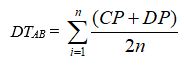
For the direct trust calculation, Control packets and Data packets are taken periodically for a given time period with the number of interactions or transactions which occurred during that time period.

Summation of CP and DP is calculated and the average trust value is taken as the direct trust.

Equations for the Direct Trust calculation:

CP = (RREQ + RPLY + HELLO + RERR) / 4

DP = NDF / NDR



DT - Direct trust value within the time period (T)

CP - Control Packets

DP - Data Packets

*n* - number of interactions or transactions happened during the time period (T)

**Procedure:** Calculate Direct Trust (DT) for time period T

1. Identify neighbour nodes
2. **for** every neighbor node do
3. get number of RREQ
4. get number of RPLY
5. get number of ERR
6. get number of HELLO
7. calculate CP for T time duration
8. get number of data packets sent within T
9. get number of data packets received within T
10. calculate DP for T time duration
11. calculate DT for T time duration
12. add DT to Trust table
13. **end for**

For each network node, trust values for their neighbor nodes are calculated and stored along with other important information in a table called Trust Tableas shown in Table 2.1.1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Neighbour node | Direct Trust  (DT) | Indirect Trust  (IDT) | Global Trust  (GT) | Interactions  (In) | Blacklist  (BL) |
|  |  |  |  |  |  |

Table 2.1.1: Trust Table

## **Indirect Trust Calculation**

The concept of a maturity level is considered here where the priority is given to nodes which have long-term relationships or a higher number of interactions with the evaluating node. And based on this maturity level a weight can be calculated for each neighbor node when considering their recommendation trust value in order to calculate the indirect trust value in the end. This concept allows nodes to give more importance to recommendations sent by long-term neighbors rather than short-term neighbors.

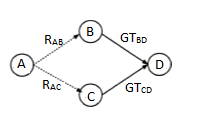


Figure 2.1: Sample Network Diagram with 4 nodes

For the ease of demonstration, 4 nodes are defined as ‘A’, ‘B’, ‘C’ and ‘D’ as shown in Figure 2.1.

*RAB* -   B recommending a trust value of D to A.                     

*GTBD*  -  Global trust between B and D.

*In*    - Interactions between the two nodes.

*WB/ WC* - Weight factors.

Indirect Trust value which A have towards D (*IDTAD*) can be calculated as follows.

*IDTAD* = *WB [R’AB \* GTBD] + WC [R’AC \* GTCD]*

Where,

*WB = R’AB / R’AB+R’AC*

*WC = R’AC / R’AB+R’AC*

*R’AB = MLB\*RAB*

*R’AC = MLC\*RAC*

*MLB = InB  / (InB + InC)*

*MLC = InC  / (InB + InC)*

RAB is assumed to be equivalent to DTBD which is Direct Trust between Node B and D. A weight factor is defined in order to make the indirect trust value between 0 and 1.

**Procedure:**

1. Get trust table entries as node\_entry\_list
2. **for** every node in node\_entry\_list do
3. calculate indirect trust value for node
4. update the indirect trust value in the trust table
5. update Global Trust for the node
6. **end for**
7. **Function** calculateIndirectTrust
8. consider the node
9. get trust table entries for the given node
10. **for** each nei\_node in node\_entry\_list do
11. calculate weight for a given node
12. calculate final recommendation trust for a given node
13. get Direct Trust and Global Trust for a given source to destination node
14. Calculate weight term and get the summation of weight
15. **end for**
16. Return summation of weight
17. .**End function**
18. **Function** calculating maturity level for a given node
19. get number of interactions for the given node
20. get trust table entries for the given node
21. **for** each nei\_node in node\_entry\_list do
22. get no of interactions for each node
23. **end for**
24. calculate maturity level
25. **End Function**
26. **Function** calculating final recommendation trust for a given node
27. calculate maturity level for a given node
28. get Direct Trust and Global Trust for a given source to destination node
29. final recommendation trust = maturity level \* DT
30. Return final recommendation trust
31. **End function**
32. **Function** calculate weight for a given node
33. calculate new recommendation trust for a given node
34. get trust table entries for the given node
35. **for** each node in node\_entry\_list do
36. Calculate new recommendation trust for a given node
37. Get the summation of new recommendation trust
38. **end for**
39. Return the ratio of new recommendation trust from the summation of new recommendation trust
40. **End Function**
41. **Function** get DT and GT for a given source to destination node
42. send TRR to the neighbor node and get DT and GT for the destination node
43. **End function**
44. get trust table entries
45. **for** each node in node\_entry\_list do
46. calculate indirect trust value for node
47. update the indirect trust value in the trust table
48. Update Global Trust
49. **end for**
50. **end**

## **Identifying Trust Levels**

After calculating the direct trust and the indirect trust of the neighbors, the global trust is calculated accordingly. And this global trust value is used to define five trust levels. Next challenge is to determine which trust level each of the neighbor nodes fall in to and by considering that trust level, the decision regarding the node can be taken throughout the network lifetime. When dividing trust levels, the default threshold values are considered as follows. If necessary, the user can change these threshold values as required.

Table 3.2: Threshold Table

|  |  |  |
| --- | --- | --- |
| Trust Level | Threshold | Meaning |
| 1 | >= TH1 | Trustworthy node |
| 2 | < TH1 and >= TH2 | Partially Trusted node |
| 3 | < TH2 and >= TH3 | Selfish node |
| 4 | < TH3 and >= TH4 | Pure Malicious node |
| 5 | < TH4 | Collaborative Malicious node |

The first category is trusted nodes which are allowed in normal routing operation and data processing is actual data forwarding and receiving. The second level is partially trusted nodes which are allowed in normal routing operation and data processing is actual data forwarding and receiving as same as trusted nodes. The third level is selfish nodes which can be allowed to take part in the normal routing operation but they will not be involved in actual data processing and their trust value will be reduced by considering their maturity level or the reputation. The fourth level is pure malicious nodes which are isolated from the network and information about these misbehaving nodes are broadcast by evaluating node to its neighbor nodes before actual route discovery commences. Therefore, those nodes are deleted from all the trust tables, recommendation tables, and backup tables. These nodes are also excluded from the route discovery process and therefore, the authentication is ensured.

Structure of the recommendation table and backup table is as follows.

Table 3.3: Recommendation Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neighbor node | Recommending nodes | Reputation Value/  Maturity level | Blacklist | Recommendation value |
|  |  |  |  |  |

Table 3.4: Backup Table

|  |  |  |  |
| --- | --- | --- | --- |
| Neighbor node | Trust Value | Time duration/ Backup time | Analyzed results |
|  |  |  |  |

Finally, the fifth trust level; collaborative malicious nodes which are similar to pure malicious nodes and for these nodes also, the same procedure of isolation and broadcasting to neighbors is done. But unlike pure malicious nodes, collaborative malicious nodes have also collaborated with one or more of its directly connected nodes in order to harm the network. In that case, apart from isolating this particular node, its neighbors who highly recommended this node to the evaluating node should also get a trust value deduction for giving higher recommendations regarding a malicious node. So, after deleting information about the collaborative malicious node, the penalty process is carried out. Once the broadcasting message is sent, the nodes who receive the message will blacklist this node until it confirms that the node is malicious for the second time.

When discovering the collaborative malicious nodes, one has to go through the Spiral Model which helps in determining pure malicious and collaborative malicious nodes. Every node can execute the spiral model over the period of time or when needed. After getting the same blacklisted node for the second time from another neighbor evaluating node can confirm the collaborative node and it will be deleted from the tables.

**Procedure:** Identify trust levels for each node.

1: Considering a node, select the set of global trust values (Z) for its neighbors from the trust table.

2: **for** every element Bi ∈ Z do

3: Bi is checked with predefined threshold value ranges

4: **if** Bi > 0.6, then

5: **if** Bi > 0.8, then

6: set Bi's trust level as 1 in the trust table.

7: **else**

8: set Bi's trust level as 2 in the trust table.

9: **end if**

10: **else**

11: **if** Bi > 0.4, then

12: set Bi's trust level as 3 in the trust table and reduce the indirect trust value from the trust table by given reduction factor.

13: **else**

14: send to spiral model to find out if the node is ‘Pure malicious’ or ‘Collaborative malicious’.

15: **end if**

16: **end if**

17: **end for**

## **Spiral Model**

As the advanced categorizing of the malicious nodes will have to go to the spiral model which have the collaborative malicious node discovery process. In the spiral model, there are three mains different phases.

### **Collaborative malicious node discovery process**

This is the phase where one does the advanced categorization for the malicious nodes and identify the collaborative malicious nodes by analyzing the dynamic behavior of the nodes. It cannot predict a collaborative using only one malicious behavior, and it must have more historical records or trust records. Mainly for this purpose, a backup table will be maintained where the recent records of the trust table will be stored and each entry on the backup table is associated with a timeout. Initially, it has predetermined a range for the trust with high trust value (HT) and low trust value (LT) and using the backup table records and current trust record it can compare the values against the time. For a given time period it can analyze the trust values, and after getting the analyzed report or plot, it can check for outliers within the given range HT – LT. If it contains any outliers or there are any unanticipated dynamic changes of the trust values it can be suspected as a collaborative malicious node. Otherwise, it can be a pure malicious node without any dynamic changing behavior. The range can be changed according to the user specification.

Figure 2.2: Flow Chart for the Spiral Model

Figure 3.1 Spiral Model

### **Penalty phase**

Once a malicious activity is detected in a node, a trust value deduction has to be done and for this purpose, a penalty phase is conducted. In cases like neighbor nodes recommending higher values for a malicious node, the penalty phase is conducted. A reduction factor is calculated based on the maturity level or the reputation of the node. Immediately after this, trust value reduction from the indirect trust value, old trust values in the trust tables should be updated with the newly calculated value. According to the updated trust value, the particular neighbor nodes should be redirected to the trust level classification phase in order to re-categorize their trust levels.

**Procedure:** collaborative malicious node discovery Algorithm (Spiral model)

1: Get the highest trust value and the lowest trust value for the node for a given time range and marked them as value boundary for outliers

2: Then compare the current trust value to verify whether it is within the range or not.

3: If the current trust value is within the range, it's categorized as a pure malicious node.

4: Then it (the node who executes this) can delete that record from all of its tables and can broadcast message to alert others.

5: Then it will terminate the pure malicious identified process.

6: If the current trust value is not within outliers it's categorized as collaborative malicious (CM) node.

7: Then it (the node who executes this) can edit its trust table blackList flag to true.

8: Identify all the neighbors of identified CM node and reduce their trust value since they have given the incorrect recommendations.

9: Broadcast to the other nodes

10: Go to the Identifying\_trust\_levels algorithm again.

### **After transmission phase**

In this phase, each neighboring node will receive details about collaborative malicious nodes through a broadcast message and the evaluating node has to check if the malicious node is in its own neighbor set. For the simplicity of the process, some notations to represent a different set of nodes were considered as below.

Collaborative malicious node CM1

Neighbor set NE

Blacklisted node set BL

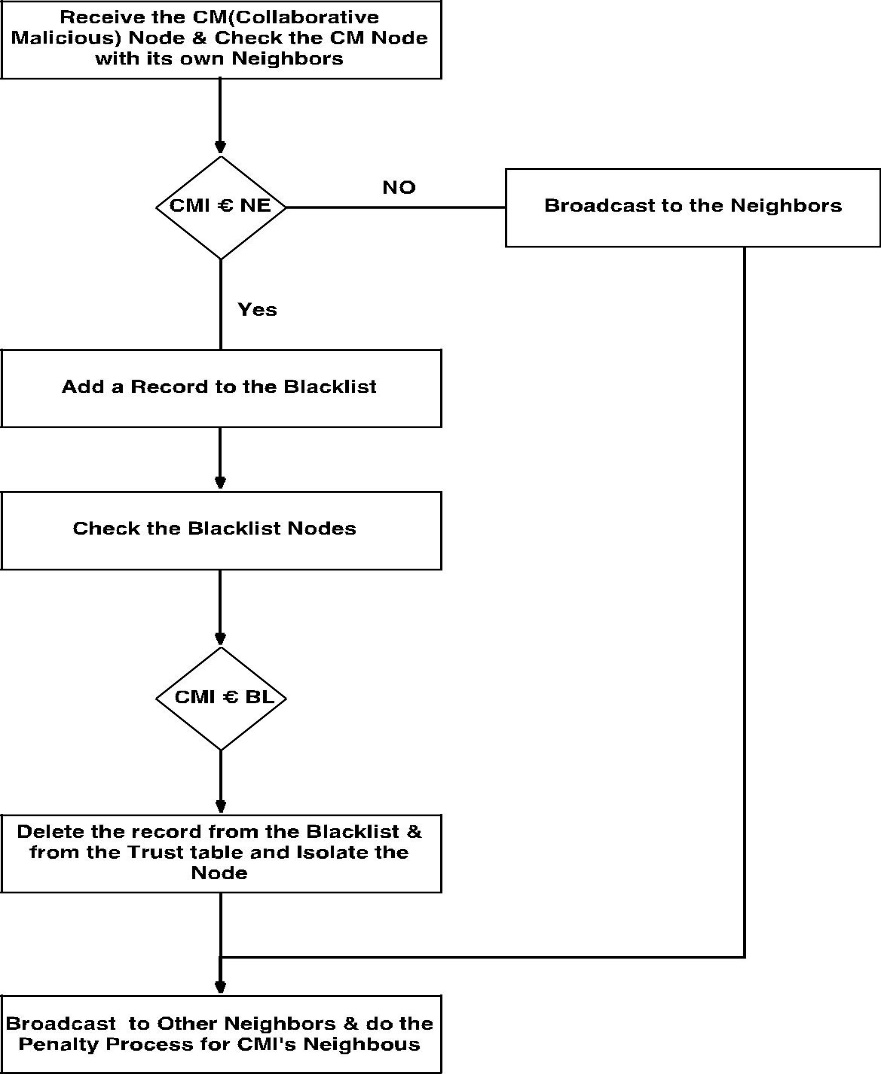


Figure 2.3: After Transmission Phase

Figure 3.2: After Transmission Phase

**Procedure:**

1: receive broadcast message and identify the p\_CM

2: **for** every element Bi ∈ BlackListed nodes do

3: **if** Bi  equals p\_CM then

4: delete p\_CM in trust table

5: delete p\_CM in recommendation table

6: **break**

7: **end if**

8: **end for**

9: **for** each neighbour node do

10: **if** p\_CM found then

11. Mark as blacklist in trust table

12. Mark as blacklist in recommendation table

13. **end if**

14. **end for**

15. Send broadcast message about p\_CM

16. **for** each node which recommended p\_CM do

17. Calculate the reduction factor

18. Update the indirect trust in the trust table

19. **end for**

20. **end**

When sending recommendation values using TRR, if the recommending node has been blacklisted by the recommender node, the evaluating node checks if it has been blacklisted on his trust table as well. If it is blacklisted, it concludes that this particular node is indeed malicious and isolates it immediately by deleting all information about it and avoiding any further interactions with the malicious node.

## **2.5. Deep Reinforcement Learning Model**

Reinforcement Learning (RL) model is trained to achieve a particular goal through the optimal path. It will assign a positive reward for correct action and negative reward for incorrect action. RL model can predict more accurate result without utilizing more historical data of the relevant scenario.

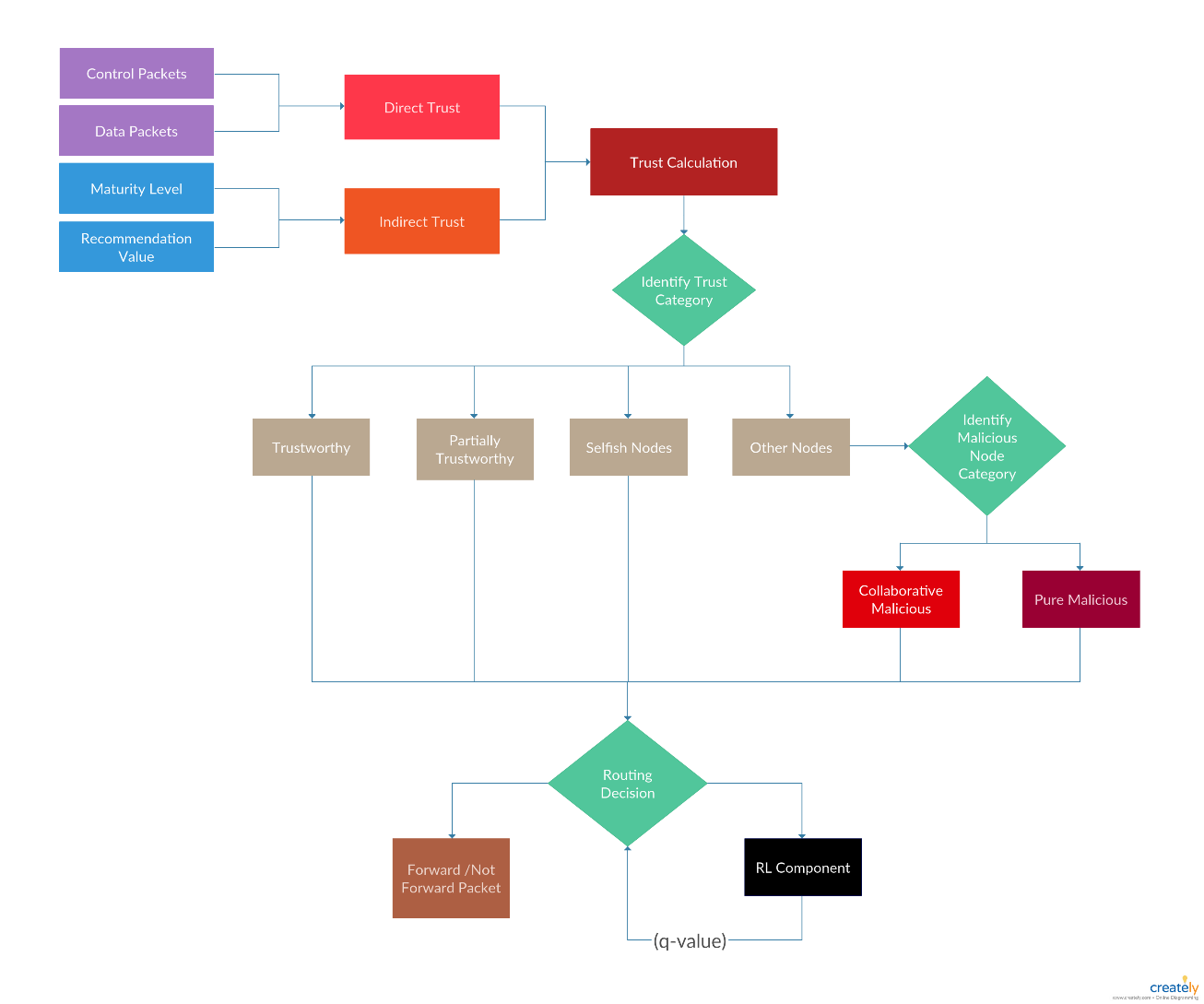


Figure 2.4: System Diagram

As in Figure 2.4, global trust value will be sent as input to the RL component. Then it will generate a q-value based on defined rewards. This q-value can determine the most trustworthy path to forward packets. If the q-value is high then it will be considered as the more trustworthy route and if q-value holds a low value then it will be an untrustworthy route.

## **2.6 Trust based routing protocols framework**

Trust-routing/trust framework alone cannot transform a protocol into a trust-based protocol. Usually, trust-based routing protocols have different favors in terms of the algorithm and trust value calculation. Therefore, the framework does provide only the abstract classes and a basic trust table. It is the responsibility of the trust framework consumer to provide a suitable trust protocol implementation to run on it. Trust framework acts as a mediator between a concrete trust-based protocol and ns-3 core modules.

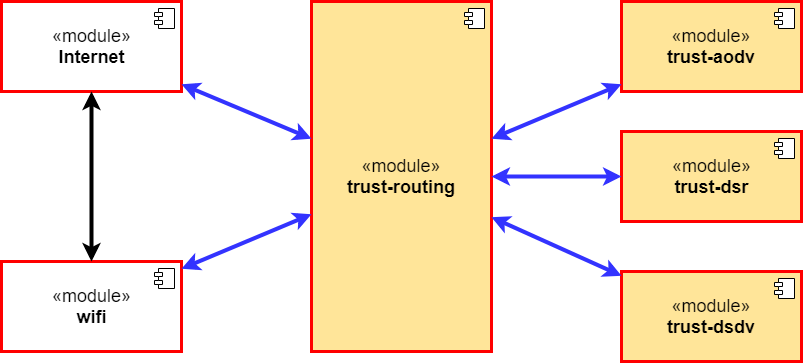


Figure 2.5: Trust Based Routing Protocol

As in the diagram, trust-aodv, trust-dsr and trust-dsdv are few of those such implementations provided to trust framework. Implementation modules contain all the different behaviors and how the entire protocol manipulation is defined. It can be stated that the trust-routing is an umbrella module that facilitates different trust implementations.

## **2.7 Performance Metrics Collection Component**

The main method of evaluating the performance of MANET is a simulation. Performance analysis will be implemented in order to compare the Trust-based routing protocol with the AODV routing protocol. The following performance metrics such as packet dropping ratio, packet delivery ratio and end to end delay are evaluated. Performance is recorded for a number of nodes including malicious nodes that are placed randomly in the network.

Performance is measured on the basis of parameters which are described as follows:

• **Packet Delivery Ratio**

Packet delivery ratio of the overall network is measured by the number of packets actually received, divided by the number of packets actually sent. If the packet delivery ratio is high, the routing protocol will be complete and more efficient. Delivery Ratio is used in most of the transmissions at a constant rate.

Packet Delivery Ratio = (Number of packets received / Number of packets sent) \* 100%

• **End to End Delay**

It is measured by the average time taken by a packet to transfer from the source to the destination. Hence it is calculated by the difference between the arrival times and sending time of packets from the source to the destination, the results will be divided by a total number of interactions. This metric is useful to understand the delay caused while discovering the path from the source to the destination.

End to End Delay = (Arrival time – Sending time) / Number of interactions

• **Packet Dropped Ratio**

Packet drop ratio is calculated by the difference between the total number of packets actually sent and the total number of packets actually received during the simulation.

Packet Drop Ratio = (Number of packets sent - Number of packets received /

Number of packets sent) \* 100%

## **2.8. Technical Objectives (S/W and H/W Requirements)**

### **2.8.1. Technologies**

C++ and Python are used as the programming languages to implement the Trust-based routing protocol.

### **2.8.2. Software Requirements**

**NS-3**

ESTAODV protocol is a new enhancement which surpasses the traditional MANET routing protocols. This new variant of the protocol has to be tested against a mobile ad-hoc network. But unfortunately, it’s not feasible to try out this new protocol only for one MANET environment. This protocol should be tested for different MANET setups for factors such as a number of nodes, the distance between nodes, mobility speed of nodes etc. Therefore, network simulations are known to be the best solution to analyze the results in such a situation.

**PyViz**

PyViz is a live simulation visualizer. It does not use any trace files. PyViz can be most useful for debugging purposes such as figuring out if mobility models are what you expect, where packets are being dropped, etc.

PyViz is used to visualize the behavior of the nodes and the transmission of the packets between neighbor nodes in the routing protocol. It will also be used to analyze the performance and behavior. PyViz has been integrated into mainline of ns-3.

**Valgrind**

Valgrind is a popular memory check library that is used to find the memory leaks when running the application. As the main protocol implementation was done in C++ context, this library can be used to utilize and eliminate the potential memory threads of ESTAODV protocol.

**TensorFlow**

Tensorflow is an open-source library developed and maintained by Google. It is used for high-performance computations which is more related to machine learning tasks. As there is a reinforcement learning component in our research, these libraries have to be used when it comes to calculating the Q-value for a given node. Tensorflow 1.8 version which is the latest version at this time is being used for this purpose. Python is used to communicate with the library and it will be used to communicate from the C++ ns-3 source code as an integration.

### **2.8.3. Hardware Requirements (Minimum)**

Memory: 4GB RAM

Disk space: 2GB of free disk space

CPU: Intel i3-2100 3.1GHz

Cache: 1.0 GHz

# **3. Results and Discussions**

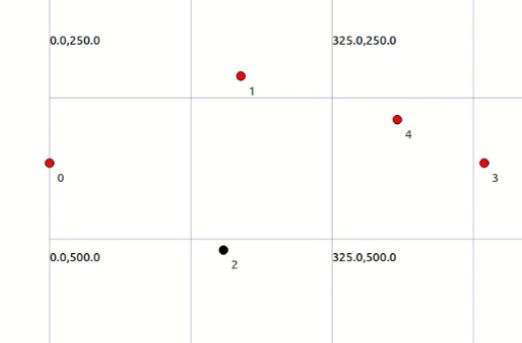
****

Figure 3.1: Network with 5 nodes

Once the network simulator is started, the each one of the tables is filled dynamically. For the ease of demonstration, a network to 5 network nodes including a malicious node (indicated in black) as shown in Figure 3.1 is considered here. Following are the trust table records printed on the terminal for each network node at the 10th second and the 90th second of network lifetime.

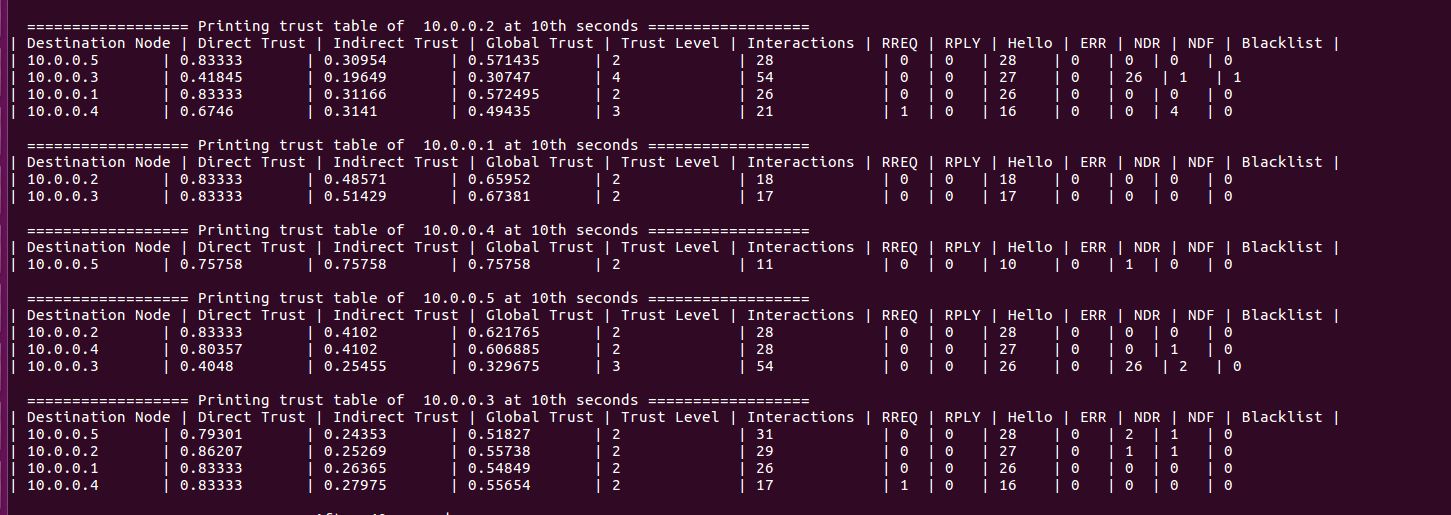


Figure 3.2: Trust tables at the 10th second

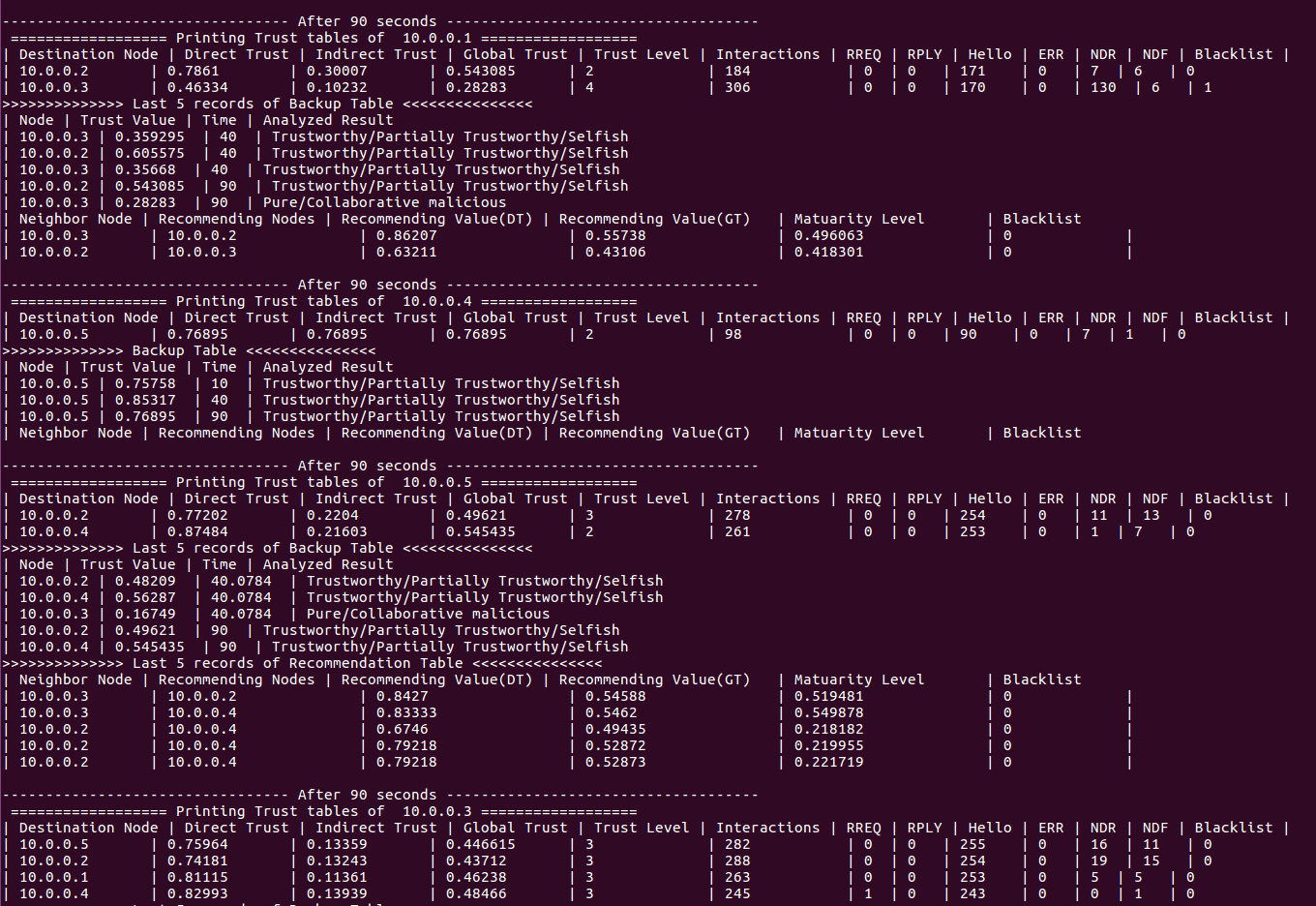


Figure 3.3: Trust tables at the 90th second

Referring to the above figures, it can be observed that all the nodes are dynamically calculating and updating their trust table records. As shown in Figure 3.2, the node 1 (with the IP address of 10.0.0.2) has recognized its neighbor, node 2 (10.0.0.3) to be pure malicious. According to the trust table of 10.0.0.2, the malicious node has been blacklisted immediately. And after this process, the information has to be broadcasted to all the nodes.

By referring to Figure 3.3, it is clear that all the nodes have received this broadcast and have deleted the node from their trust table hence the malicious node is isolated from the network. None of the nodes no longer has any information regarding the node 10.0.0.3.

Once a recommendation is received, information regarding those recommendation values is stored in the recommendations table of each node as well. The maturity value calculated has been changed over time with the increasing number of interactions between the nodes. This has affected indirect trust and therefore, the global trust has been changed in the trust tables as well.

The performance analysis is implemented to compare trust-based routing protocol with AODV routing protocol based on 3 matrices in Network Simulator 3 (NS3). 20 nodes are placed for the simulation with the random waypoint mobility model.

Table 3.1.: Simulation Parameters

|  |  |
| --- | --- |
| **System Parameters** | **Values Utilized** |
| Number of Mobile Nodes | 20 |
| Number of malicious nodes | 0,2, 4, 6, 8 |
| Mobility Model | Random Waypoint Mobility |
| Simulation Duration | 100 Sec |

## **3.1. Vary the number of malicious nodes**

Since identifying and isolating malicious nodes is the main aim of the simulation, malicious nodes are included in the network to validate the performance. The number of malicious nodes is increased step by step and analyzed using the results in AODV routing protocol and trust-based routing protocol.

### **3.1.1. Packet Delivery Ratio**

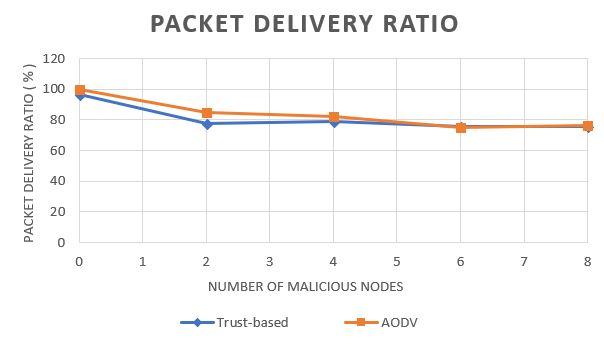


Figure 3.4: Packet Delivery Ratio

Packet delivery ratio of the trust-based model varies with the AODV routing protocol. The packet delivery ratio of both trust-based routing protocol and AODV routing protocol is decreased when the number of malicious nodes increases. The malicious nodes act as forwarders until trustworthy routes are established. Because of that, the packet delivery ratio is decreased.

### **3.1.2. End to End Delay**

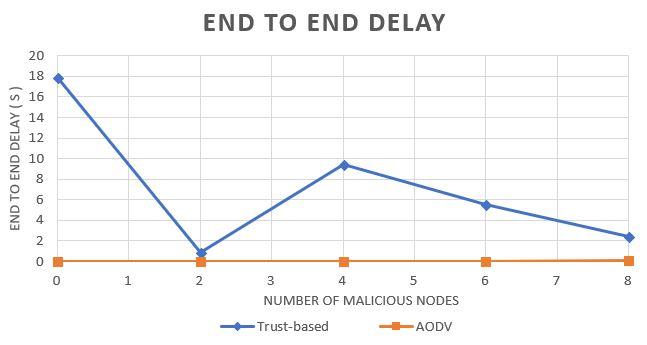


Figure 3.5: End to End Delay

End to end delay of trust-based routing protocol is high compared to AODV routing protocol. In the trust-based routing protocol, there is a trust calculation process to be executed which consumes time. Because of that trust-based model has a higher delay sum than traditional AODV.

### **3.1.3. Packet Drop Ratio**

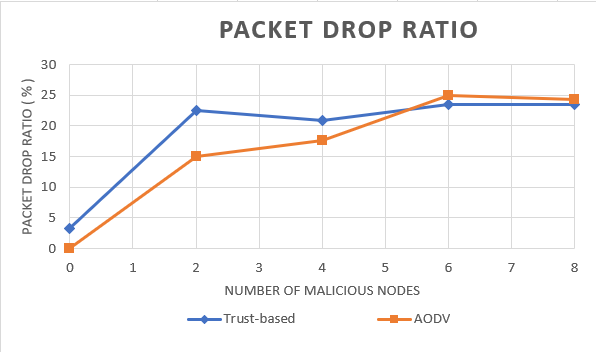


Figure 3.6: Packet Drop Ratio

Packet drop ratio of trust-based routing protocol and AODV routing protocol are slightly increased when the number of malicious nodes increases. When the number of malicious node increases, trust-based routing protocol shows lower packet drop ratio than the AODV routing protocol.

## **3.2. Malicious Node Detection Accuracy**

Results are recorded to analyze the detection accuracy of malicious nodes in the trust-based routing protocol. Both pure malicious and collaborative malicious nodes are considered in the ‘Malicious’ category.

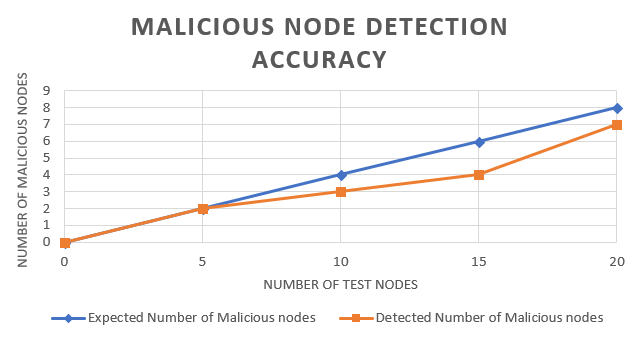


Figure 3.7.: Malicious node detection accuracy

Malicious node detection rate has increased when the number of test nodes in the network gets increased. The number of malicious nodes is constant while incrementing a number of nodes. The above figure ensures that the trust-based routing protocol detects malicious nodes in the MANET with higher accuracy and shows accurate results.

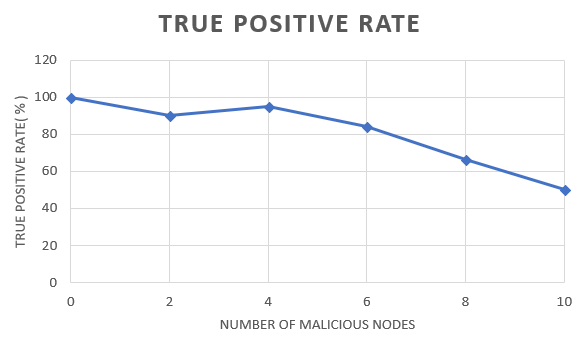


Figure 3.8: True positive rate

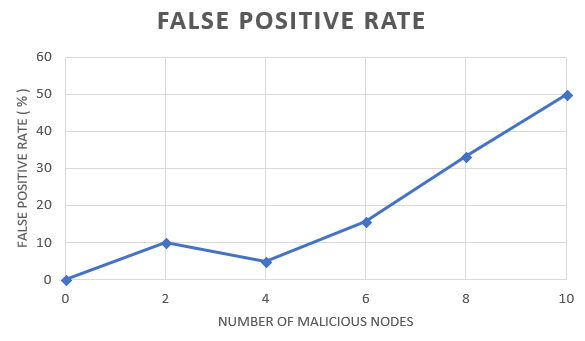


Figure 3.9: False positive rate

Trust-based routing protocol shows the results for identifying malicious nodes with a high true positive rate and a low false positive rate. The above figures represent the true positive rate and false positive rate when incrementing a number of malicious nodes while keeping the total number of nodes as a constant. There as a result of increasing the density of malicious nodes, troublesome of detecting malicious nodes through the trust-based model is increased.

# **SUMMARY OF CONTRIBUTION**

|  |  |  |
| --- | --- | --- |
| **Member** | **Component** | **Task** |
| M.A.J. Niroshan | Trust Recommendation Request Protocol (TRR) | * Define trust recommendation request protocol which will be used to request recommendation values from other network nodes within the network. * Efficient trust recommendation request packet processing and transmission. |
| Trust Based Framework | * Generic trust-based framework which can be plugged for other routing protocols. * Enable trust framework to support for Ipv4 routing protocols. * Enable trust framework to support for Ipv6 routing protocols. * A new module in ns-3 to demonstrate the usage of trust-based protocols framework. |
| K.A.H. Kodithuwakku | Evaluate Indirect Trust | * Implement maturity level calculation algorithm considering the interactions between neighbor nodes. * Implement recommendation model to keep track of recommending nodes and their recommendation values and current status. * Implement a weight factor to be used for the indirect trust calculation process. * Implement a reduction factor for misbehaving nodes and to update their trust value according to the penalty. |
| Trust Level classification | * Define and categorize nodes, based on their behavior using the threshold values defined. |
| Liyanage S.C.G. | Trust Level Identification | * Implement Spiral model which can distinguish collaborative malicious nodes and pure malicious nodes. |
| Deep Reinforcement Learning | * Define a reinforcement learning (RL) model. * Implement Q-learning mechanism to calculate Q-value for each node. * Analyze the Q-value and make predictions to detect the secured and reliable routes. |
| M.A.S.H. Kularatne | Evaluate Direct Trust | * Implement Control Packets calculation considering RREQ, RPLY, ERR, HELLO packets. * Implement Data Packets calculation considering number of packets forwarded and number of packets received. * Implement Direct Trust calculation algorithm considering control packets, data packets and the interactions between neighbor nodes within an exact time period. * Generate the trust table and store Direct Trust value for the neighbor nodes. |
| Performance Metrics Collection Component | * Perform network simulations using the visualizer. * Visualize the behavior of the nodes and the transmission of the packets between neighbor nodes |

# **CONCLUSION**

In this research, a trust-based framework is developed to investigate the dynamic behavior of network nodes and to find the most secure route for packet transmission instead of the shortest path in the traditional AODV. Since the first version of traditional AODV had security vulnerabilities which posed serious threats to data packets transmitted via MANETs, different approaches to secure AODV have been considered such as calculating both direct trust and indirect trust for nodes and dividing malicious nodes into two categories; pure malicious and collaborative malicious. The concept of relationship maturity based on the interaction count between two nodes and also a trust recommendation model is introduced to take recommendations from neighbor nodes regarding a particular node. Based on direct trust and indirect trust, a global trust value is calculated to evaluate the trustworthiness of nodes and to categorize them into five trust levels. A performance analysis is implemented to compare the trust-based framework with traditional AODV routing protocol as well. Although this framework is to detect the most trustworthy path, one could ideally work to do further research in the above-mentioned domain and come up with a different solution.

# **REFERENCES**

[1]     S. S, P. K, and T. A, “Authentication Using Trust to Detect Misbehaving Nodes in Mobile Ad hoc Networks Using Q-Learning,” *Int. J. Netw. Secur. Its Appl.*, vol. 8, no. 3, pp. 47–64, 2016.

[2]     K. J. R. Liu, “Information theoretic framework of trust modeling and evaluation for ad hoc networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 305–317, 2006.

[3]       R. Dalal, “Different Ways to Achieve Trust in MANET,” *Int. J. AdHoc Netw. Syst.*, vol. 2, no. 2, pp. 53–64, 2012.

[4]       C. Fountas, “Swarm Intelligence: The Ant Paradigm,” Springer, Berlin, Heidelberg, 2010, pp. 137–157.

[5]     P. Velloso, R. Laufer, D. De O. Cunha, O. C. Duarte, and G. Pujolle, “Trust management in mobile ad hoc networks using a scalable maturity-based model,” *IEEE Trans. Netw. Serv. Manag.*, vol. 7, no. 3, pp. 172–185, 2010.

[6]    T. Farid and A. Prahladachar, “Secure Routing with AODV Protocol for Mobile Ad Hoc Networks.”

[7] S. Subramanian and B. Ramachandran, “QOS Assestion in MANET Routing based on Trusted AODV,” Int. J. Ad hoc, Sens. Ubiquitous Comput., vol. 3, no. 3, pp. 135–143, 2012.

[8]   J. S. S.Uma, “Human Interaction Pattern Mining Using Enhanced Artificial Bee Colony Algorithm S.,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 3, no. 10, pp. 10131–10138, 2015.

[9] R. K. Bar, J. K. Mandal, and M. M. Singh, “QoS of MANet Through Trust based AODV Routing Protocol by Exclusion of Black Hole Attack,” *Procedia Technol.*, vol. 10, pp. 530–537, 2013.

[10]   Y. Yoo and D. P. Agrawal, “Why does it pay to be selfish in a MANET?,” *IEEE Wirel. Commun.*, vol. 13, no. 6, pp. 87–97, 2006.

[11] R. Neogy, C. Chowdhury, and S. Neogy, “Reliability of mobile agents for reliable service discovery protocol in MANET,” *arXiv Prepr. arXiv1111.1865*, vol. 3, no. 5, pp. 229–243, 2011.

[12] G. Santhi, A. Nachiappan, M. Z. Ibrahime, R. Raghunadhane, and M. K. Favas, “Q-learning based adaptive QoS routing protocol for MANETs,” *Int. Conf. Recent Trends Inf. Technol. ICRTIT 2011*, pp. 1233–1238, 2011.