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

Project ID: 041

Project Proposal Report

M.A.J. Niroshan

K.A.H. Kodithuwakku

Liyanage S.C.G.

M.A.S.H. Kularathna

Bachelor of Science Special (Honors) in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2018

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

Project ID: 041

Project Proposal Report

(Proposal documentation submitted in partial fulfilment of the requirement for the Degree of Bachelor of Science Special (honors)

In Information Technology)

Bachelor of Science Special (Honors) in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 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. Communication within a network is a crucial operation as it has a major impact on 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 packets to transmit among nodes within a small period of time. It is possible that some nodes might not behave as we expect. It can eventually cause to a considerable amount of packet drops. It shows that existing security mechanisms have failed to distinguish trustworthy nodes and malicious nodes. Usually, 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. 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 MANET networks.

# **TABLE OF CONTENTS**

[**DECLARATION** iii](#_gjdgxs)

[**ABSTRACT** iv](#_30j0zll)

[**TABLE OF CONTENTS** v](#_1fob9te)

[**LIST OF FIGURES** vii](#_3znysh7)

[**LIST OF TABLES** vii](#_2et92p0)

[**LIST OF ABBREVIATIONS** vii](#_tyjcwt)

[**LIST OF APPENDICES** viii](#_1t3h5sf)

[**1**](#_4d34og8) **INTRODUCTION** 1

[**1.1**](#_2s8eyo1) **Background** 1

[**1.2**](#_17dp8vu) **Literature Review** 2

[**1.2.1**](#_3rdcrjn) **Black hole attack in AODV routing protocol [1]** 2

[**1.2.2**](#_26in1rg) **Authentication using trust to detect misbehaving nodes in mobile ad Hoc networks using Q-Learning [1]** 3

[**1.2.3**](#_lnxbz9) **Information theoretic framework of trust modeling and evaluation for ad hoc networks [2]** 3

[**1.2.4**](#_35nkun2) **Different ways to achieve trust in MANET [3]** 5

[**1.2.5**](#_1ksv4uv) **Trust management in mobile ad hoc networks using a scalable maturity-based model [5]** 6

[**1.2.6**](#_44sinio) **Secure routing with AODV protocol for mobile ad hoc networks [6]** 7

[**1.2.7**](#_2jxsxqh) **QoS assertion in manet routing based on trusted AODV (ST-AODV) [7]** 7

[**1.2.8**](#_z337ya) **EBoX: Evidence of behavior information exchange mechanism against selfish attacks [8]** 8

[**1.2.9**](#_3j2qqm3) **QoS of MANET through trust based AODV routing protocol by exclusion of black hole attack [9]** 8

[**1.3**](#_1y810tw) **Research Gap and Research Problems** 8

[**2**](#_4i7ojhp) **OBJECTIVES** 10

[**2.1**](#_2xcytpi) **Main Objectives** 10

[**2.2**](#_1ci93xb) **Specific Objectives** 10

[**3**](#_3whwml4) **RESEARCH METHODOLOGY** 11

[**3.1**](#_2bn6wsx) **Direct Trust Calculation** 11

[**3.2**](#_qsh70q) **Indirect Trust Calculation** 13

[**3.3**](#_3as4poj) **Identifying Trust Levels** 18

[**3.4**](#_1pxezwc) **Spiral Model** 21

[**3.4.1**](#_49x2ik5) **Collaborative malicious node discovery process** 21

[**3.4.2**](#_2p2csry) **Penalty phase** 23

[**3.4.3**](#_147n2zr) **After transmission phase** 24

[**3.5**](#_3o7alnk) **System Diagram** 27

[**3.6**](#_23ckvvd) **Gantt Chart** 28

[**4**](#_ihv636) **DESCRIPTION OF PERSONAL AND FACILITIES** 29

[**5**](#_32hioqz) **REFERENCES** 31

[**6**](#_1hmsyys) **APPENDICES** 32

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
|  |  | Page |
| Figure 1.1 | Sample network diagram with 3 network nodes | 4 |
| Figure 1.2 | Node A receives recommendations about Node D [5] | 6 |
| Figure 2.1 | Sample Network Diagram with 4 nodes | 14 |
| Figure 3.1 | Spiral Model | 22 |
| Figure 3.2 | After Transmission Phase | 25 |
| Figure 3.3 | System Diagram | 27 |
| Figure 3.4 | Gantt chart | 28 |

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
|  |  | Page |
| Table 3.1 | Trust Table | 13 |
| Table 3.2 | Threshold Table | 18 |
| Table 3.3 | Recommendation Table | 19 |
| Table 3.4 | Backup Table | 19 |

# **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 |

**LIST OF APPENDICES**

|  |  |  |
| --- | --- | --- |
| Appendix | Description | Page |
| Appendix-A | Gantt Chart | 28 |

# **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 occur 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. Most of the times flexibility and accessibility of a wireless network is 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, 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. Simply the packets might be 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 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 level.

## **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 ‘Black hole attack’. When 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 drop 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 into a higher value.

Apart from RREQ and RPLY, there are 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 those 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]**

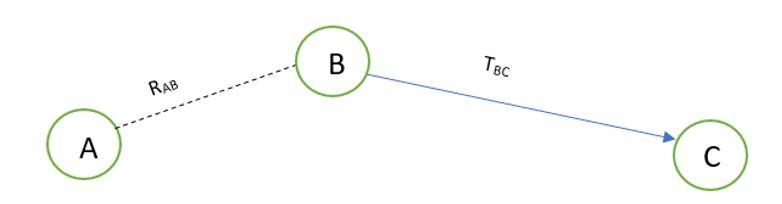
Authentication which is the key factor to be considered in MANET can be categorized into two sections called pre-authentication and post-authentication. As the name denotes pre-authentication is initial network deployment and post-authentication is mechanism to detect nodes in the network over a period 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 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 unwise 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. Afterwards, 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 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 entropy-based model they have come up with an equation to calculate TABC which is same as the indirect trust between node A and C.





TABC = RABTBC

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 introduced as TRR (Trust Recommendation Request) to get the trust value of a particular node by requesting from the other neighbor nodes.

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 they are attempting to detect malicious nodes.

One drawback of this solution is that malicious nodes can collaboratively provide wrong recommendations for other nodes. Therefore, a mechanism should be required to detect collaborative malicious nodes. By analyzing this past history of network node interactions, we came up with 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.

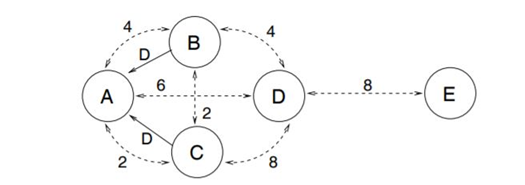
### **Different ways to achieve trust in MANET [3]**

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 swarm intelligence paradigm [4] to model the protocol-based schema. Communication among network nodes happened through agents similar to ants [3] in ABED. Ants collected information which is called as pheromones [3]. Based on pheromones, ants find optimal path for measure trust evidence. Generalized Reputation Evidence (GRE) [3], is another instance for protocol-based scheme.
* System level-based scheme - Under system level-based, it will give rewards to trustworthy nodes and give 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 selfish nodes and malicious nodes.
* Cluster based scheme
* Maturity based scheme
* PKI (Public Key Infrastructure) based scheme

They have not yet implemented a way to further categorize malicious nodes.

### **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. Simply, a network node will broadcast messages to its directly connected nodes (neighbor nodes) requesting for the recommendation value regarding a particular neighbor node.



The 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 of 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 is occurred due to invalid cryptographic information. Intrusion Detection Model (IDM) and Intrusion Response Model (IRM) comes into front as the solution. IDM uses neighbor node information to detect misbehaving nodes. If misbehavior count is greater than threshold value, it will broadcast about 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

There Qr, Qp and Qd are the intermediate values to calculate 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 to 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 ignore 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 which their ability to forward data packets and forward RREQ for a given network node. Finally trust value is calculated as a multiplication of forwarded data packets ratio and forwarded RREQ packets ratio. Then trust value is recorded in the routing table to take 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 as 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 behaviors of nodes. This can lead to many IoT device failure with 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 network nodes that brings harm. There is no way to prove complete trustworthiness is only depend on direct interactions among each node in the network. There might have chances of getting high accuracy for trust values by getting recommendations from other network nodes. At the same time could not come to a better decision only depending on indirect trust value. That will arise the requirement of calculating the average value of direct trust value and indirect trust value when taking a better conclusion on trustworthiness of nodes. On the other hand, definition for trust among the network nodes is similar to trust among human beings. Direct trust is the trust which builds with the experience among each other. When someone has suspects about that trust, going to 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 about collaborative behaviors of malicious nodes. Sometimes 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 misbehaving as a team in the network. Therefore, it is important to distinguish the type of malicious nodes.

# **OBJECTIVES**

## **Main Objectives**

* Evaluate trustworthiness of nodes in ad hoc network and categorize according to defined trust levels in the ad hoc network.
* Detect malicious nodes and isolate them from the network.
* Analyze performance metrics when detecting the secured and reliable routes using proposed trust-based schema.

## **Specific Objectives**

* Calculate direct trust and indirect trust values to measure the trustworthiness of network nodes.
* Distinguish pure malicious nodes and collaborative malicious nodes in the ad hoc network.
* Implement Trust Recommendation Request (TRR) algorithm to retrieve recommendations from other directly connected network nodes.
* Predict the secured routes in MANET using Q-Learning mechanism.

# **RESEARCH METHODOLOGY**

## **Direct Trust Calculation**

Over the period of time, a node wants to send a packet to a particular destination. According to our model, initially, all the nodes broadcast the HELLO packets instead of initiating route discovery process or checking their own routing table for the desired route. So that every node ensures it’s one hop neighboring nodes ultimately only one hop neighbors respond to the hello packets because they are in same communication range. From that, every node can conclude how many nodes are staying as one-hop neighbors. In AODV the following control packets are used. In route discovery, route request (RREQ), route reply (RPLY) packets are used. Route error (RERR) and HELLO packets are used in route maintenance process. 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 but utilize probability of such packets are relatively low compared with well-behaving nodes.

Where NDF denotes the number of data packets actually forwarded, and NDR denotes the number of packets actually received over 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 getting the CP (Control Packet) and the DP (Data Packet), we can calculate the direct trust (DT). Specially here we are not going to consider the Control packets and Data packets for only one transaction for calculating the direct trust, instead of that periodically within a given time period we have to get the summation of CP and DP and finally get the average trust value as the direct trust. It is given by,

where DT is direct trust value within time duration T, CP is Control Packets for T time duration, DP is Data Packets for T time duration and *n* is number of interactions or transactions for T time duration.

**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

BEGIN

FOR each neighbor node do

Get number of RREQ

Get number of RPLY

Get number of ERR

Get number of HELLO

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

Sent = number of Data Packets Sent

Received = number of Data Packets Received

DP = sent data packets / received data packets

Save DT in trust table DT to Trust table

END FOR

END

For all nodes, we calculate the trust for their neighbor nodes and store those information in a table called Trust Tableas follows.

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

Table 3.1: Trust Table

## **Indirect Trust Calculation**

Here we consider the concept of maturity level same as the human-related maturity where we give priority to node which is having long-term relationship or the interactions for a long time with the evaluating node and later on based on this maturity level of each and every node we can give a weight without getting the recommendation trust value as it is. This concept allows nodes to give more importance to recommendations sent by long-term neighbors rather than short-term neighbors.

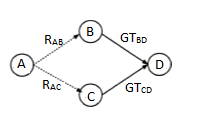


Figure 2.1: Sample Network Diagram with 4 nodes

For the ease of demonstration, we will define 4 nodes as ‘A’, ‘B’, ‘C’ and ‘D’ as 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 records for 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)*

We assume that RAB is 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. Endfunction
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

Pseudo code for Indirect Trust

BEGIN

    Function calculate indirect trust value for a given node

        Pass In: in\_node

        node\_entry\_list = getTrustTableEntries()

        w\_sum = 0

        FOR each nei\_node in node\_entry\_list DO

            W = calculate W for a given node

            R'\_nei\_node = calculating R' for a given node

            rec[] = get DT and GT for a given source to destination node

            cal\_w\_term = W \* (R'\_nei\_node \* rec[1])

            w\_sum = w\_sum + cal\_w\_term

        END FOR

Pass Out: w\_sum

    Endfunction

    Function calculating maturity level for a given node

    Pass In: p\_node

        Intr\_p\_node = p\_node.getInteractions()

        node\_entry\_list = getTrustTableEntries()

        In\_all = 0

        FOR each entry in node\_entry\_list DO

            In\_all = In\_all + entry.getInteractions()

        END FOR

        ML = Intr\_p\_node / In\_all

    Pass Out: ML

    Endfunction

    Function calculating R' for a given node

    Pass In: p\_node

            ML = calculating maturity level for a given node

            rec[] = get DT and GT for a given source to destination node

            DT = rec[0]

            R' = ML \* DT

    Pass Out:  R'

    Endfunction

    Function calculating W for a given node

    Pass In: p\_node

            R'\_nei\_node = calculating R' for a given node

            R'\_all = 0

            node\_entry\_list = getTrustTableEntries()

            FOR each node in node\_entry\_list DO

                R'\_node = calculating R' for a given node

                R'\_all = R'\_all + R'\_node

            END FOR

            W = R'\_nei\_node / R'\_all

    Pass Out: W

    Endfunction

   Function get DT and GT for a given source to destination node

    Pass In: nei\_node, in\_node

        rec[] =  sendTRR(nei\_node, in\_node)        //both DT and GT stored inside rec

    Pass Out: rec[]

    Endfunction

    node\_entry\_list = getTrustTableEntries()

    FOR each node in node\_entry\_list DO

        calculate indirect trust value for node

        update the indirect trust value in the trust table

        updateGlobalTrust(node)

    END FOR

END

## **Identifying Trust Levels**

After calculating the direct trust and the indirect trust of the neighbors, we are getting the global trust and based on the global trust value we have defined five trust levels. Next challenge is to identify the trust level of the neighbors, and by considering the trust level, we have to get the routing decision. According to the trust value, five trust levels are listed down in below table. When we are dividing the trust level, we have to consider a threshold value for each and every level.

|  |  |  |
| --- | --- | --- |
| 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 |

Table 3.2: Threshold Table

These values are classified into five categories, first one is trusted nodes; we can allow those nodes in normal routing operation and data processing is actual data forwarding and receiving, second is partially trusted nodes; we can allow those nodes in normal routing operation and data processing is actual data forwarding and receiving same as trusted nodes, third level is selfish nodes; we allow those nodes 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, next level is, pure malicious nodes; those nodes are isolated from the network and information about the misbehaving nodes can be broadcast by evaluating nodes before actual route discovery commences. Therefore, those nodes are deleted from all the trust tables, recommendation tables, and backup tables. By the way, such nodes are isolated from route discovery process, and therefore authentication is ensured by excluding those nodes.

Structure of the recommendation table and backup table as follows.

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

Table 3.3: Recommendation Table

|  |  |  |  |
| --- | --- | --- | --- |
| Neighbour node | Trust Value | Time duration/ Backup time | Analysed results |
|  |  |  |  |

Table 3.4: Backup Table

Finally, we have the collaborative malicious nodes; same as the malicious nodes for these nodes also do the same process and take the same actions and apart from that special action will be taken for the collaborative malicious nodes such as other than broadcasting and isolation of the node particular node will be added to the blacklist and declare it as a blacklisted node. After blacklisting the node, it will broadcast and notify the neighboring nodes in order to collaboratively convey the collaborative malicious behavior of the particular node and take the decision about the node. When discovering the collaborative nodes, we have to go through the Spiral model which we discuss in next section. Every node can execute the spiral model over the period of time or when needed. After getting the same blacklisted node 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 the 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**

BEGIN

Node A select the set of global trust values (Z) for its neighbors from the trust table.

FOR every element Bi ∈ Z DO

IF Bi> 0.6

IF Bi>0.8

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

ELSE IF

Set Bi's trust level as 2 in the trust table.

END IF

ELSE

IF Bi>0.4

Set Bi's trust level as 3 in the trust table.

Reduce the indirect trust value from the trust table by the given reduction factor.

ELSE

Send to spiral model to find out if Bi is ‘pure malicious’ or ‘collaborative malicious’.

END IF

END IF

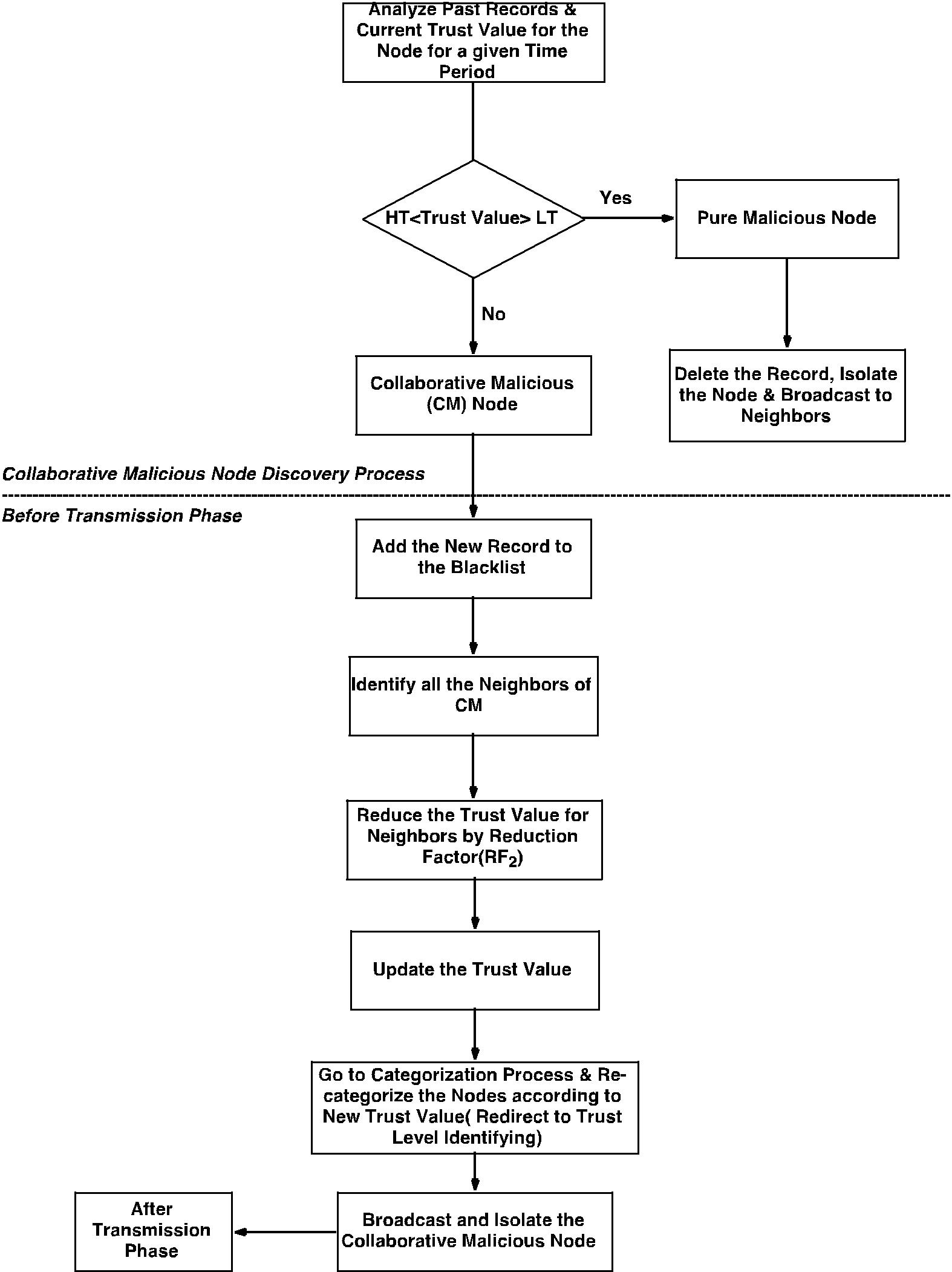
END FOR

END

## **Spiral Model**

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

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

This is the phase where we do the advanced categorization for the malicious nodes and identify the collaborative malicious nodes by analyzing the dynamic behavior of the nodes. Only using one record we cannot predict a collaborative malicious behavior, and we have to have more historical records or trust records. For this purpose, mainly, we are maintaining a backup table where we store the recent records of the trust table and each entry on the backup table is associated with a timeout. Initially, we have predetermined range for the trust with high trust value (HT) and low trust value (LT) and using the backup table records and current trust record we can compare the values against the time. For a given time period we can analyze the trust values, and after getting the analyzed report or plot, we can check for outliers within the given range HT – LT. If it contains any outliers or there are any sudden dynamic changes of the trust values we can suspect it 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. 



### **Penalty phase**

Same as the trust level identification phase here also we reduced the trust value of the particular recommending node with the help of a reduction factor. Reduction factor will be calculated based on the maturity level or the reputation of the node. Immediately after the trust reduction, old trust value in the trust table 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 identification phase in order to re-categorize their trust levels.

The node which is evaluating will wait to get another recommendation from one of its neighbor for the same blacklisted node as a collaborative malicious node to confirm it as a collaborative node. That process will discuss under the After-Transmission phase.

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

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

2: Then compare current trust value is in between the range or not.

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

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

5: So that will terminate the pure malicious identified process.

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

7: Then it (the node who execute 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.

BEGIN

p\_M =passed-in malicious node

IF trust value is not an outlier

THEN

Delete from trust table

Send Broadcast to delete node

ELSE

Mark node as blacklist in trust table

FOR each node which recommended p\_M DO

Calculate reduce factor

Recalculate indirect trust

Update global trust

END FOR

Broadcast neighbors about p\_M node

GOTO Identifying\_trust\_levels

END IF

END

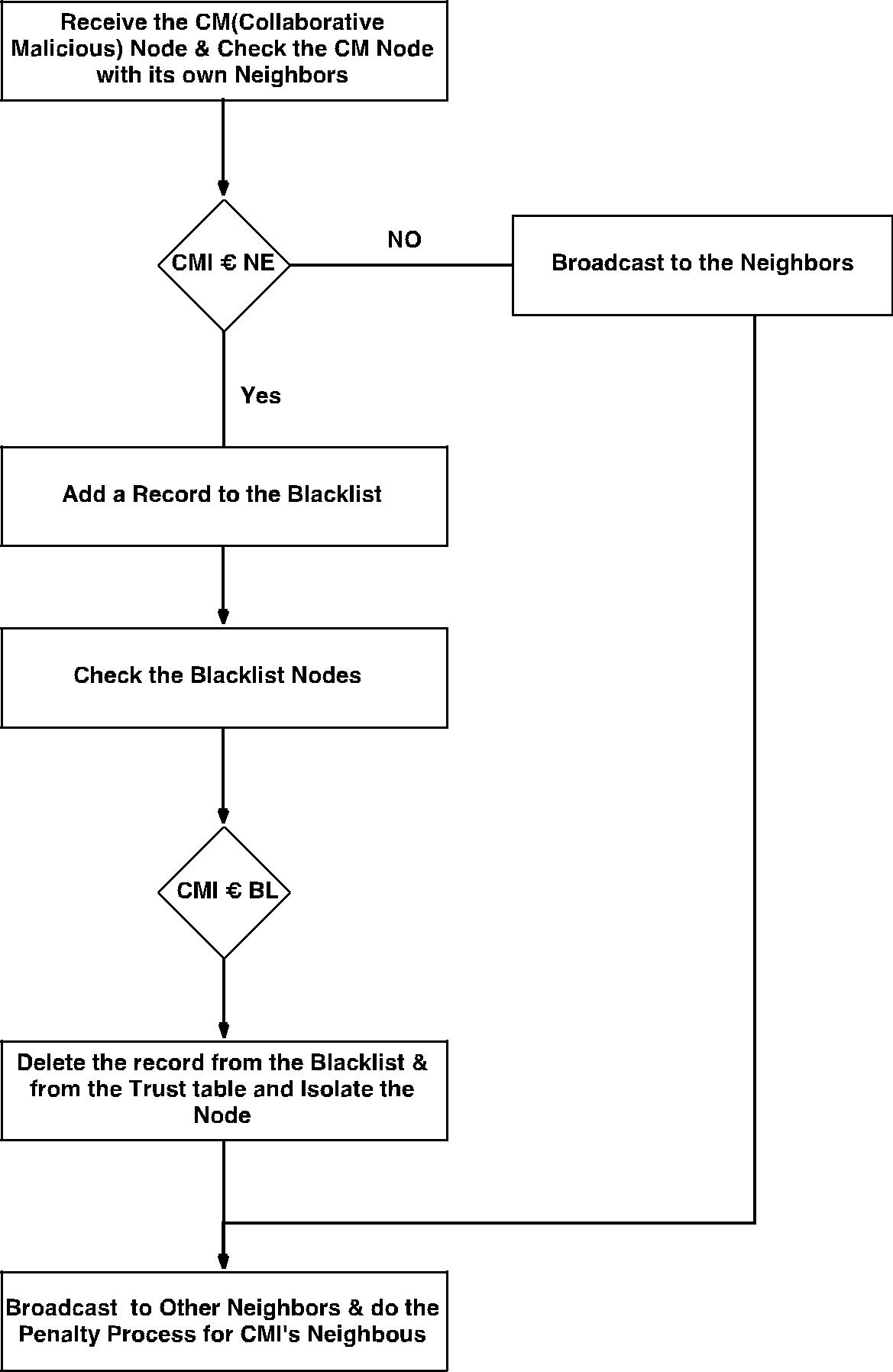
### **After transmission phase**

In this phase, each neighboring node will receive the collaborative node detail through the broadcast, and it has to check its own neighbor set. For the simplicity of the process, we consider some notations to represent a different set of nodes as below.

Collaborative malicious node CM1

Neighbor set NE

Blacklisted node set BL



**Procedure:**

1: read the received 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

BEGIN

FOR every element Bi ∈ BlackListed nodes do

IF p\_CM equals Bi then

Delete p\_CM in Trust Table

Delete p\_CM in Recommendation Table

BREAK;

END IF

END FOR

FOR each neighbor node do

IF **p\_CM** found then

Mark as black list in trust table

Mark as black list in recommendation table

END IF

END FOR

Send broadcast message about **p\_CM**

FOR each node which recommended **p\_CM** do

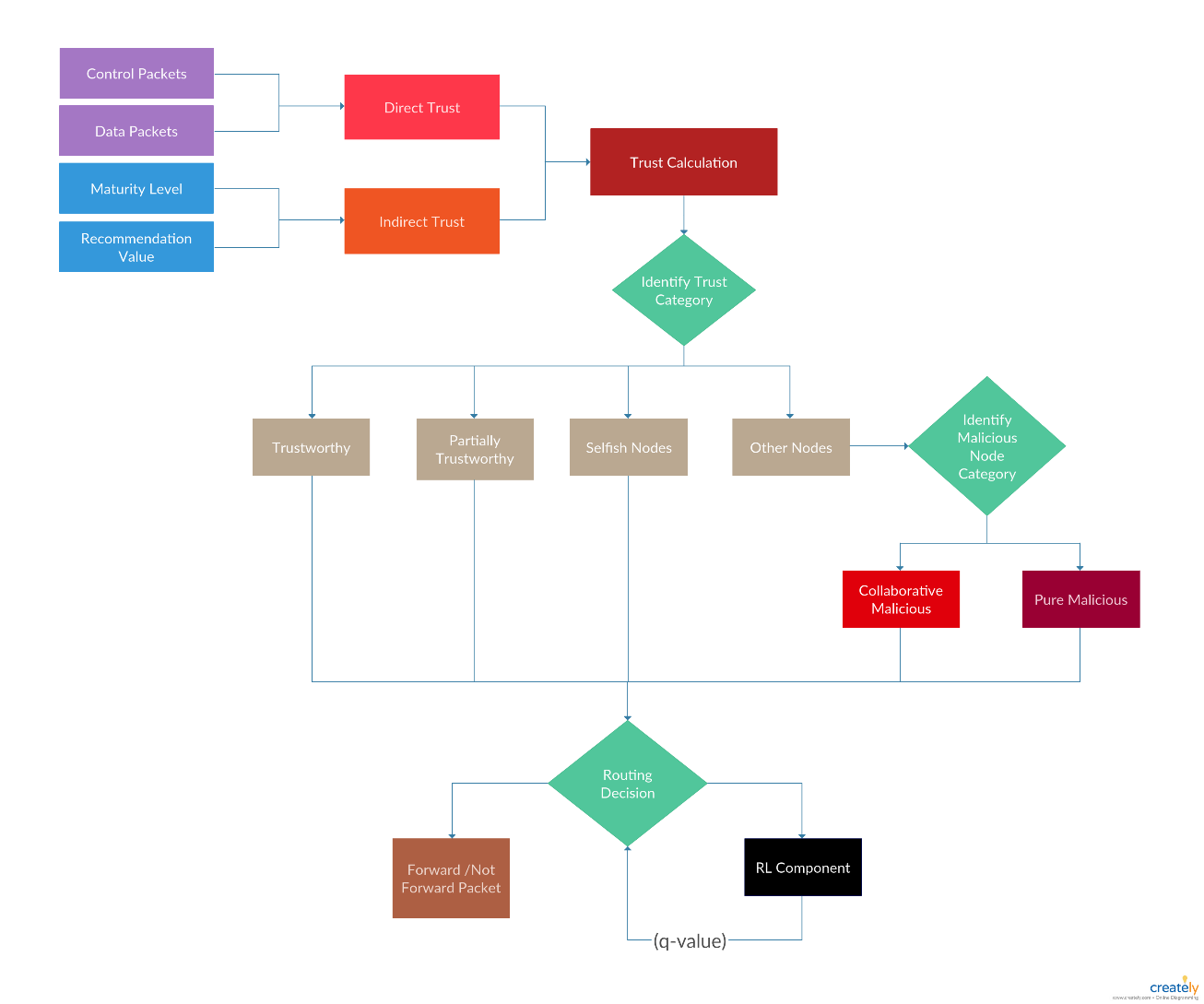
Calculate a reduction factor

Recalculate indirect trust in trust table

END FOR

END

## **System Diagram**

Figure 3.3: System Diagram

## **Gantt Chart**

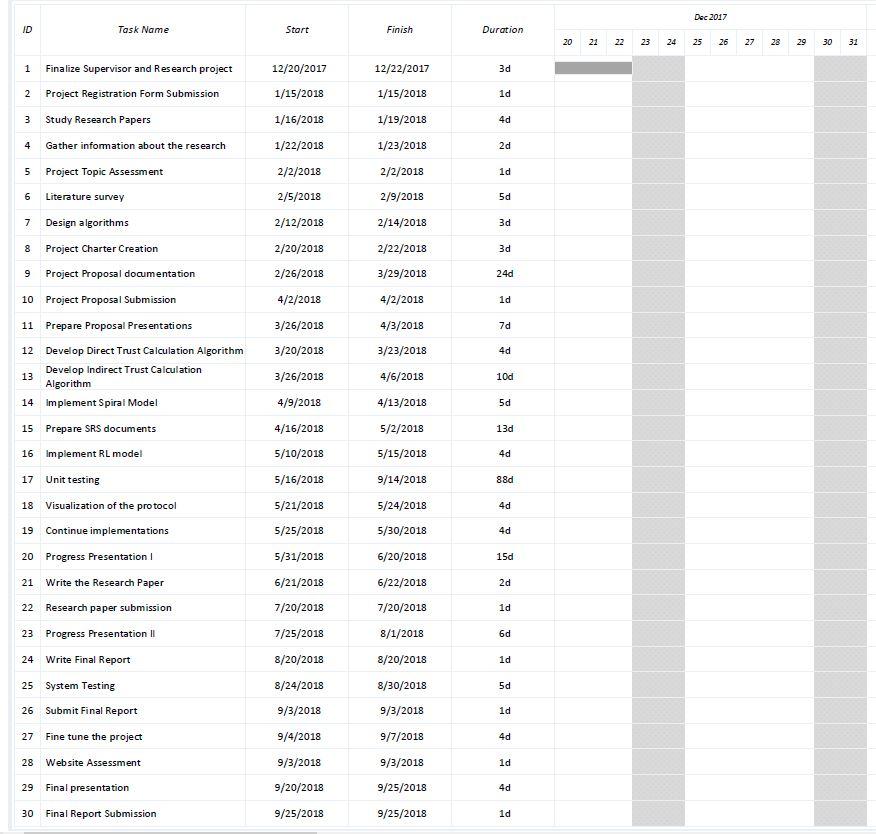


Figure 3.4: Gantt chart

# **DESCRIPTION OF PERSONAL AND FACILITIES**

|  |  |  |
| --- | --- | --- |
| **Member** | **Component** | **Task** |
| M.A.J. Niroshan | Trust Recommendation Request Protocol (TRR) | * Define trust recommendation request protocol which can be used to request recommendation values from other nodes within the network. * Efficient trust recommendation request packet processing and transmission |
| Trust Based Framework | * Generic trust framework which can be plugged for other routing protocols. * Enable framework to support for Ipv4 routing protocols * Enable framework to support for Ipv6 routing protocols |
| 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 indirect trust calculation process. * Implement a reduction factor for misbehaving nodes and to update their trust value according to the penalty. |
| Trust Level Identification | * 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 be capable of distinguishing collaborative malicious nodes and pure malicious nodes. |
| Deep Reinforcement Learning | * Define a reinforcement learning (RL) model to generate a reward for inputted global trust value. * Implement Q-learning mechanism to calculate Q-Value for each node. * Analyse the Q-value and make predictions to detect the secured and reliable routes. |
| M.A.S.H. Kularathna | 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 behaviour of the nodes and the transmission of the packets between neighbour nodes. |

# **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.

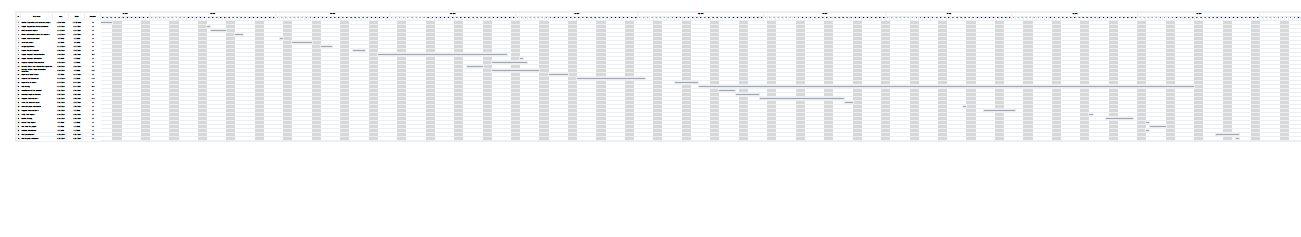
[9] R. K. Bar, J. K. Mandal, and M. M. Singh, “QoS of MANet Through Trust based AODV Routing Protocol by Exclusion of Black Hole Attack,” *Procedia Technol.*, vol. 10, pp. 530–537, 2013.

[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.

1. **APPENDICES**



Appendix - A