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

Project ID: 18-024



Preliminary Progress Review (PPR)

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

# DECLARATION

We declare that this is our own work and this document does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or an 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 |  |  |

**TABLE OF CONTENTS**

**Declaration….………………………………………………………………………….**

**Table of Contents…...………………………………………………………………….**

**List of Figures………………………………………………………………………….**

**List of Tables………..…………………………………………………………………..**

**List of Abbreviations…………………………………………………………………..**

**List of Appendices………………………………………………………………….…..**

**1. Introduction**

**1.1.         Purpose…………………………………………………………**

**1.2.         Scope……………………………………………………………**

**1.3.         References (Place at the end of the report)..............................**

**1.4.         Overview…………………………………………………….….**

**2. Statement of the work**

**2.1.    Background information and overview…………………………**

**2.2. Identification and significance of the problem………………..**

**2.3.    Technical objectives........................................................................**

**2.4.    Detail design (Technical approach)...............................................**

**2.5.    Sources for test data & analysis………………………………….**

**2.6. Anticipated benefits……………………………………………..**

**3. Project plan or schedule……………………………………………………..**

**4. Research constraints………………………………………………………….**

**5. Specified deliverables………………………………………………………..**

**6. References……………………………………………………………………..**

**List of Figures**

**Figure [2.1] Sample network diagram with 3 network nodes………………………….**

**Figure [2.2]: Node A receives recommendations about Node D.....................................**

**Figure [2.3] Sample Network Diagram with 4 nodes........................................................**

**Figure [2.4]:Trust level classification process flow...........................................................**

**Figure [2.5]: Collaborative malicious node discovery process flow................................**

**Figure [2.6]: After transmission phase flow......................................................................**

**Figure [2.7]: Trust Recommendation Request Protocol................................................**

**Figure [2.8]: Overall system diagram................................................................................**

**Figure [2.9]: Network nodes position 1 ...........................................................................**

**Figure [2.10]: Network nodes position 2............................................................................**

**Figure [2.11]: Network nodes position 3 ..........................................................................**

**Figure [3.1]: Work breakdown structure.........................................................................**

**List of Tables**

**Table [2.1] Trust Table.....................................................................................................**

**Table [2.2] Threshold Table ............................................................................................**

**Table [2.3] Recommendation Table ...............................................................................**

**Table [2.4] Backup Table ...............................................................................................**

# 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 |
|  |  |
| 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 |  |

# 

# 1. Introduction

## 1.1. Purpose

The main purpose of preliminary project report (PPR) is to understand the project architecture and to capture the quality and the performance of functional elements which are to be implemented. The current state of the project and to what extent the proposed solution is planning on moving forward from the current implementation is also addressed in this document. The significance and the uniqueness of the idea, technical and scientific challenges which might arise during the implementation are also discussed through the PPR.

This documentation is a collection of written information and graphical information that is used to convey design intent to parties that are involved in the project such as the owners. Through PPR, design and performance requirements which cannot be shown by graphical representations are demonstrated. Design solution for each individual element of the system and the extent of their relationships are described using this documentation.

**1.2. Scope**

This documentation intends on covering main aspects of the research. The main goals, objectives, possible users types and challenges that can be arised in the implementation phase are also discussed here. It is also discussing about the significance and the uniqueness of the problem that is being addressed, approaches taken to solve the problem and technical requirements to achieve them and possible limitations. Background information and an overview of the previous work are also covered based on a literature survey.

The scope of the project is segmented into fields of direct trust calculation, indirect trust calculation, identifying trust levels, collaborative malicious node discovery phase/spiral model, after transmission phase, trust recommendation request protocol and deep reinforcement learning model.

Test data, data collection procedures and the data analysis methods to be used are discussed here as well. Anticipated benefits from the proposed solution and the project plan which will give an overview of the time frame within the various parts of the project to be completed are also discussed in the document. Furthermore, it is discussing about the facts such as improving security, reliability and performance.

//research outcome

## 1.3. Overview

Mobile Ad Hoc Network (MANET) is a self organizing collection of wireless mobile nodes that form a temporary and dynamic wireless network without any infrastructure. 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. 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. 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.

To identify malicious nodes, need to evaluate trustworthiness of the nodes in ad hoc network and categorize according to defined trust levels in the ad hoc network. Direct trust and indirect trust values are used to measure the trustworthiness of the network nodes. Direct trust is the trust which builds with the experience among directly connected nodes. When a node takes recommendations regarding a node from others, it is considered as the indirect trust. In the indirect trust calculation process, Trust Recommendation Request (TRR) algorithm is implemented to retrieve recommendations from other directly connected network nodes.

After detecting malicious nodes, need to distinguish pure malicious nodes and collaborative malicious nodes in the ad hoc network. Malicious nodes should be isolated from the network and broadcast it to the neighbor nodes. Performance metrics are analyzed when detecting the secured and reliable routes using proposed trust-based schema. Reinforcement learning (RL) model is defined to generate a reward for inputted global trust value. Q-learning mechanism is used to calculate Q-Value for each node and predict the secured routes in MANET. The implemented trust-based routing protocol will be released with NS3 therefore the developers and researchers can make use of it. Trust enabled ipv4/ipv6 enhancement will be done.

# 2. Statement of the work

## 2.1. Background information and overview of previous work based on literature survey

### **2.1.1. 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.

### **2.1.2. 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] and various kinds of 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.

### **2.1.3. 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 2.1.2 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.

  
  
  
**Figure [2.1]: Sample network diagram with 3 network nodes**

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.

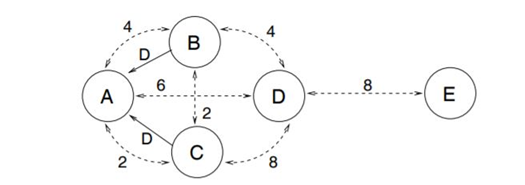
### **2.1.4. Different ways to achieve trust in MANET [3]**

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.

### **2.1.5. 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.

**Figure [2.2]: Node A receives recommendations about Node D [5]**

The Figure 2.2 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.

### **2.1.6. 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.

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

In order to increase Quality of Service (QoS) [11] 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.

### **2.1.8. 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.

### **2.1.9. 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.

## 2.2. Identification and significance of the problem

*Comment: How new or unique is the idea? How significant is the scientific and/or technical challenge?*

Nodes in mobile ad-hoc networks (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.

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. As revealed in 2.1, there already exist some trust based algorithms regarding MANET. But all these models are based either on the direct trust or the indirect trust aspects. Direct trust based models such as 2.1.2, only considers the direct trust aspect is 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 the complete trustworthiness in a network where the nodes only considers the direct interactions among each other. Considering indirect aspects such as getting trust value recommendations from neighbor nodes will result higher accuracy when it comes to evaluating trust towards a node. Indirect trust based models only considers the indirect aspects for calculating trustworthiness. But the direct interactions among nodes play a major part when it comes to trust in a network. So the proposed model will take both these factors in to when calculating trustworthiness of nodes.

According to 2.1.3, collaborative behaviors of malicious nodes aren’t considered for the trust calculation process. 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.

Although the model 2.1.8 has mentioned about selfish nodes, all the other models mentioned above in 2.1, categorizes nodes as either malicious or trustworthy. This might not be enough to represent all the types of node behaviors in the network. That is why the proposed solution has introduced a trust level classification model. This model categorizes network nodes in to five categories according to their dynamic behavior throughout the network’s lifetime and treat each category of nodes in a particular manner. This also helps reducing computation costs since now instead of having to go through each and every one of the trust values, the specific trust level of a node could be considered. Identification of different malicious node types is not an easy task. So after putting nodes in to ‘trustworthy’, ‘partially trustworthy’ and ‘selfish’ categories, the remaining nodes are sent to a separate model called ‘spiral model’ to further categorize as ‘pure malicious’ or ‘collaborative malicious’. Once a node is concluded to be not trustworthy, according to their harmfulness, a reduction factor will be calculated. This reduction factor will then be deducted from the trust value of the neighbor node.

In ‘EBoX’ model mentioned in 2.1.8, selfish nodes and misbehaving nodes are identified. And a reputation based schema is introduced in order to award credit points or deduct penalty points from the trust value which one node has towards another neighbor node. When such a node is detected, it is isolated from the network. So that in this method, the fate of a network node depends entirely on the network nodes which decides to award either penalty points or credit points. But these other network nodes might even be harmful nodes at some points and in such cases it is not fair to take a decision depending only on such an awarding. This is why the proposed solution considers a maturity model and a weight factor for the recommendation value calculation process. When a recommendation value is receives from one node, maturity model decides how matured this particular node is based on their interaction count with neighbor nodes. Then using a weight factor, the weight of the recommendation value is control according to the node’s trustworthiness and maturity level in order to calculate the indirect trust.

## 2.3. Technical objectives (specify s/w and h/w requirements)

**2.3.1. Software Requirements**

**NS-3**

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

NS-3 is a discrete-event network simulator, targeted primarily for research and educational use. It is free software, licensed under the GNU GPLv2 license, and is publicly available for researches and development purposes. NS-3 is built C++ and Python using scripting capabilities. NS-3 is used as the network simulation software to analyze the behavior of this new MANET protocol.

**PyViz**

PyViz is a live simulation visualizer. It doesn’t 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 mostly written in Python but it works both with Python and pure C++ simulations.

PyViz will be 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. Knowledge about PyGTK and GooCanvas are also required to work with pyViz.

**MATLAB**

MATLAB is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python. The main method of evaluating the performance of MANETs is simulation. The analysis of the simulation of AODV routing protocol the developed trust-based routing protocol will be done on the basis of performance metrics using MATLAB.

**TensorFlow**

Tensorflow is an open-source library developed and maintained by Google. It is used for high performance computations which is more related with machine learning tasks. As we have a reinforcement learning component in our research, we have to use these library when it comes to calculating the Q-value for a given node. We are using Tensorflow 1.8 version which is the latest version at this time. 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.

**2.3.2. Hardware Requirements (Minimum)**

Memory: 4GB RAM

Disk space: 2GB of free disk space

CPU: Intel i3-2100 3.1GHz

Cache: 1.0 GHz

## 2.4. Detail design (Technical approach)

### **2.4.1. 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**

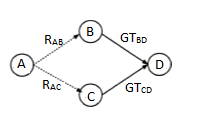
For all nodes, we calculate the trust for their neighbor nodes and store those information in a table called Trust Tableas follows.

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

Table 2.1: Trust Table

### **2.4.2. 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 [2.3]: 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 2.3.

*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. **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**

### **2.4.3. 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 2.2: 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.

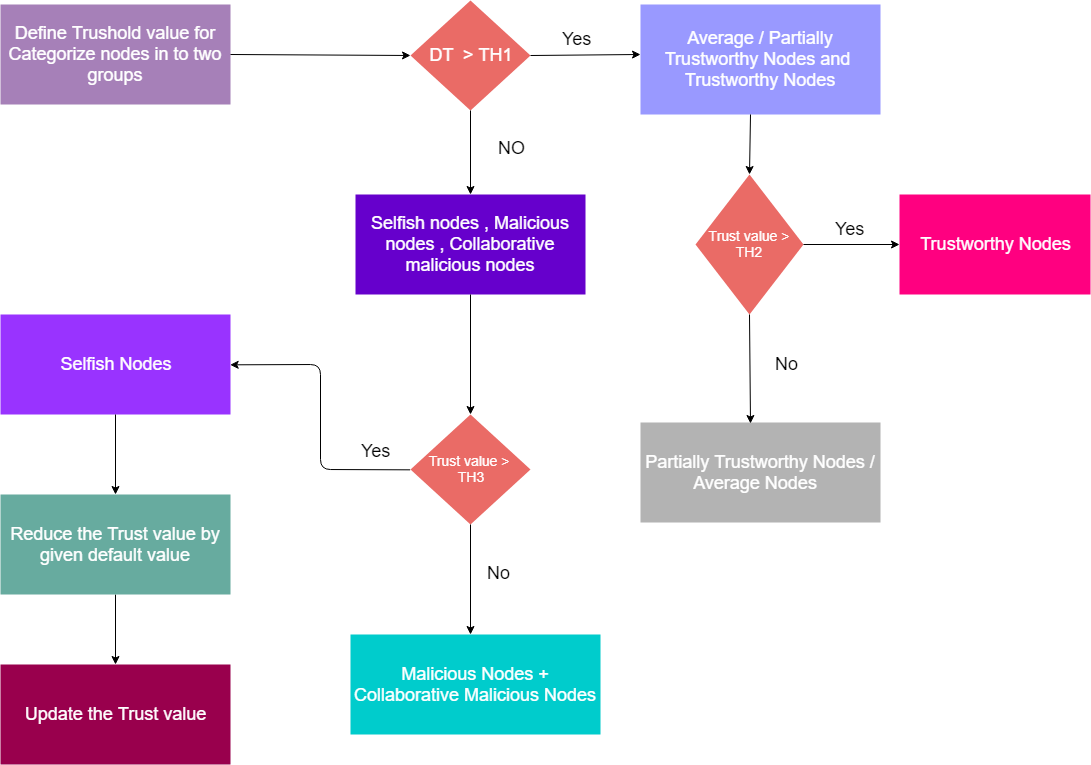
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neighbour node | Recommending nodes | Reputation Value/  Maturity level | Blacklist | Recommendation value |
|  |  |  |  |  |

Table 2.3: Recommendation Table

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

Table 2.4: 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.



**Figure [2.4]: Trust level classification process flow**

**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**

### 

### **2.4.4. Collaborative malicious node discovery process (Spiral Model)**

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.



**Figure [2.5]: Collaborative malicious node discovery process flow**

### 

### **2.4.5. 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.

**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.

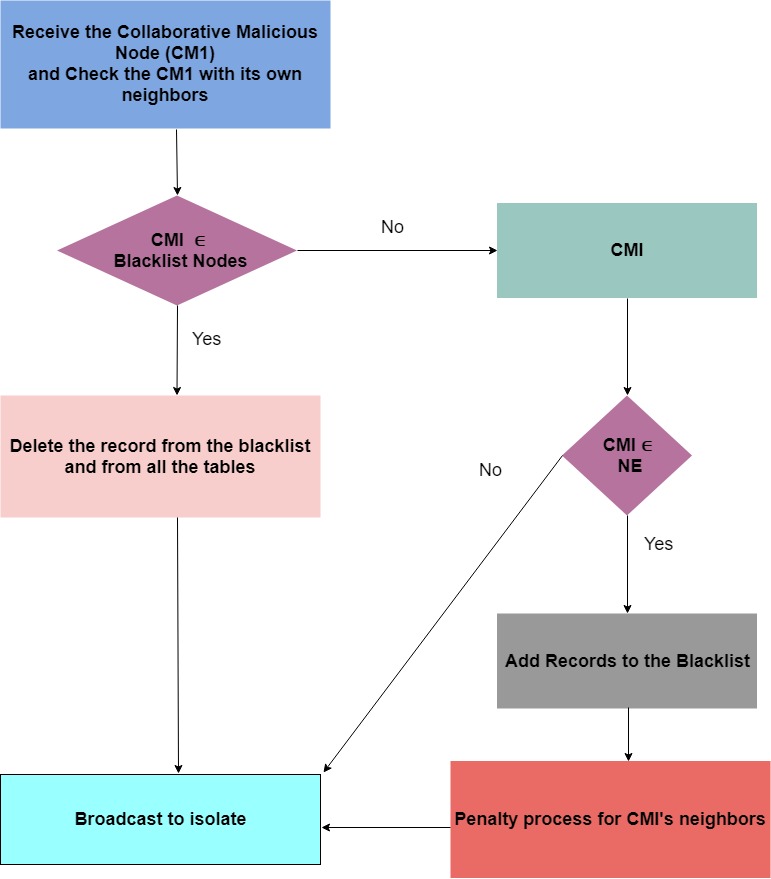
### **2.4.6. After transmission phase**

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



**Figure [2.6]: After transmission phase flow**

**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

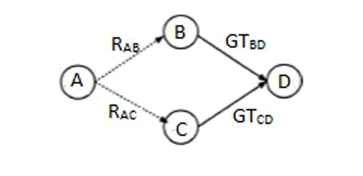
17. Calculate the reduction factor

18. Update the indirect trust in the trust table

19. **end for**

20. **end**

### **2.4.7. Trust Recommendation Request (TRR) Protocol**



**Figure [2.7]: Trust Recommendation Request Protocol**

In order to calculate indirect trust, we need to take recommendations from other network nodes. That is basically done via Trust Recommendation Request (TRR) protocol. As in the Figure [2.7], if node A needs to know the direct trust and global trust values of node D, then node A will broadcast a packet called TRR by requesting trust values of node D. Since node B and node C are directly connected to node D, they have entries in their trust tables related to node D. Therefore, they can reply back to that TRR packet.

### **2.4.8. Trust enabled framework**

//Google Summer of Code (GSoC) thing

### **2.4.9. 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 [2.8]: Overall system diagram**

As in the above diagram global trust value will be inputted to 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 consider as more trustworthy route and if q-value holds a law value then it will be an untrustworthy route.

## 2.5. Sources for test data & analysis

**Data Collection**

Our proposed trust-based framework will be applied to MANET networks. This protocol enhancement has to be validated against different types of networks. Due to different capabilities of each device, it is not feasible to setup a physical network and try out the simulation variations. Therefore, a network simulator software will be used to simulate the different types of networks.

NS-3 is a discrete-event network simulator, targeted primarily for research and educational use and it has been chosen as the network simulator for this project. It is an open source software so that it can be used for free. ‘NS-3’ can be used to gather the required network statistical data for model comparisons.

**Data Analysis**

For analysis purposes, ‘Scikit Learn’ machine learning library will be used. Using this library, it is easy to plot the data graphs and visualize the differences. ‘MATLAB’ will be used to analyze performance metrics. ‘Pyviz’, also known as the ‘python visualizer’ is a visualization tool for network simulations and demonstration purposes. This will also be used for the visualization of network activities. Refer to Figure [2.9], Figure [2.10], Figure [2.11] for the demonstration of networks activities using Pyviz.

## 2.6. Anticipated benefits

### **2.6.1. Improved Security**

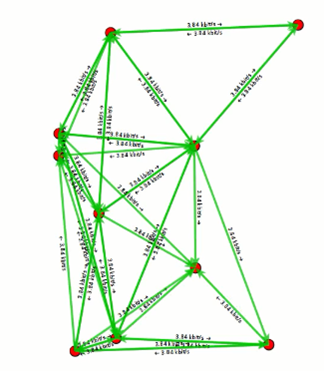
**Network attacks**

Wireless mobile networks such as MANET process a considerably huge amount of data in a small amount of time. Due to that reason these mobile network nodes has a higher throughput. In such cases, the network is prone to network attacks such as black hole attacks, wormhole attacks etc. With the proposed mechanism, the nodes who could participate in these types of attacks are identified and immediately removed from the network.

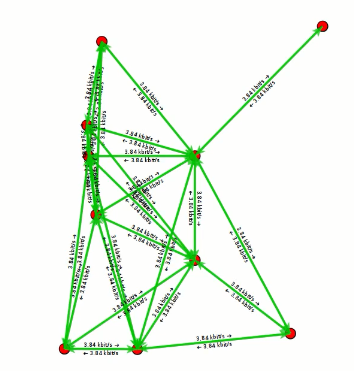
**Malicious Nodes**

With the suggested model, malicious nodes are categorized in to three types as selfish nodes, pure malicious nodes and collaborative malicious nodes. Selfish nodes are the nodes whose behavior depends on its own benefits. If the transaction of the packet benefits the selfish node, it proceeds with the transaction. Otherwise it drops the packet. In such cases packet loss ratio is increased. The other types of malicious nodes could harm the information which the packets are carrying, before passing it on to the next node. Among these, pure malicious nodes are almost always harmful and the collaborative malicious nodes might only collaborate with other neighbor nodes in order to cause harm. By categorizing these nodes in to different trust levels, different categories of nodes can be treated separately according to the manner of their harmful behavior. In order to determine the trust level of each node in the network, their behavior is being monitored frequently. By this approach we can recognize the harmful nodes before they cause major harm to the network and through that ensure the network security.

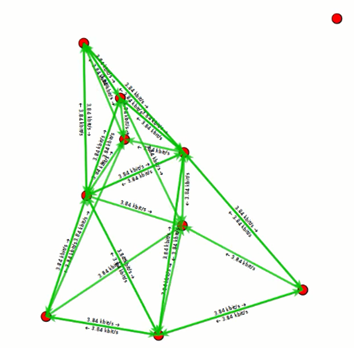
### **2.6.2. Improved Reliability**



**Figure [2.9]: Network nodes position 1**



**Figure [2.10]: Network nodes position 2**



**Figure [2.11]: Network nodes position 3**

The reliability of ad-hoc networks depends on the cooperation and trust among distributed nodes. MANET has a limited coverage area and since the nodes in these networks are mobile and not located in a specific area, they might sometimes go beyond the limited coverage area of the network. In such cases the network cannot rely on such nodes. From the figure [2.6.1.1], a regular MANET is shown with transactions happening between nodes. Figure [2.6.1.2] shows the same network after some time and one node seems to travel away from the rest of the network nodes. And figure [2.6.1.3] indicates that the particular node mentioned above has completely removed itself from the network. With the proposed mechanism, such behaviors in nodes are identified early and those nodes are treated as malicious nodes. Doing so, the rest of the network will not have greater dependencies with those nodes and reliability will not be compromised.

### **2.6.3. Lesser Performance Decrements**

Trying to perfectly solve each and every one of the issues that might occur in the network could bring the network performance down. The proposed mechanism is aware of this issue and it is made so that the performance decrements are prevented as much as possible. Categorization of nodes can be considered to show this approach. If each and every one of the nodes in the network were to be considered separately in order to evaluating their trustworthiness, each time this process happens it would take a lot of time and cost for the calculations. Since they are categorized in the proposed model, they can be evaluated according to their trust level resulting a much lesser cost.

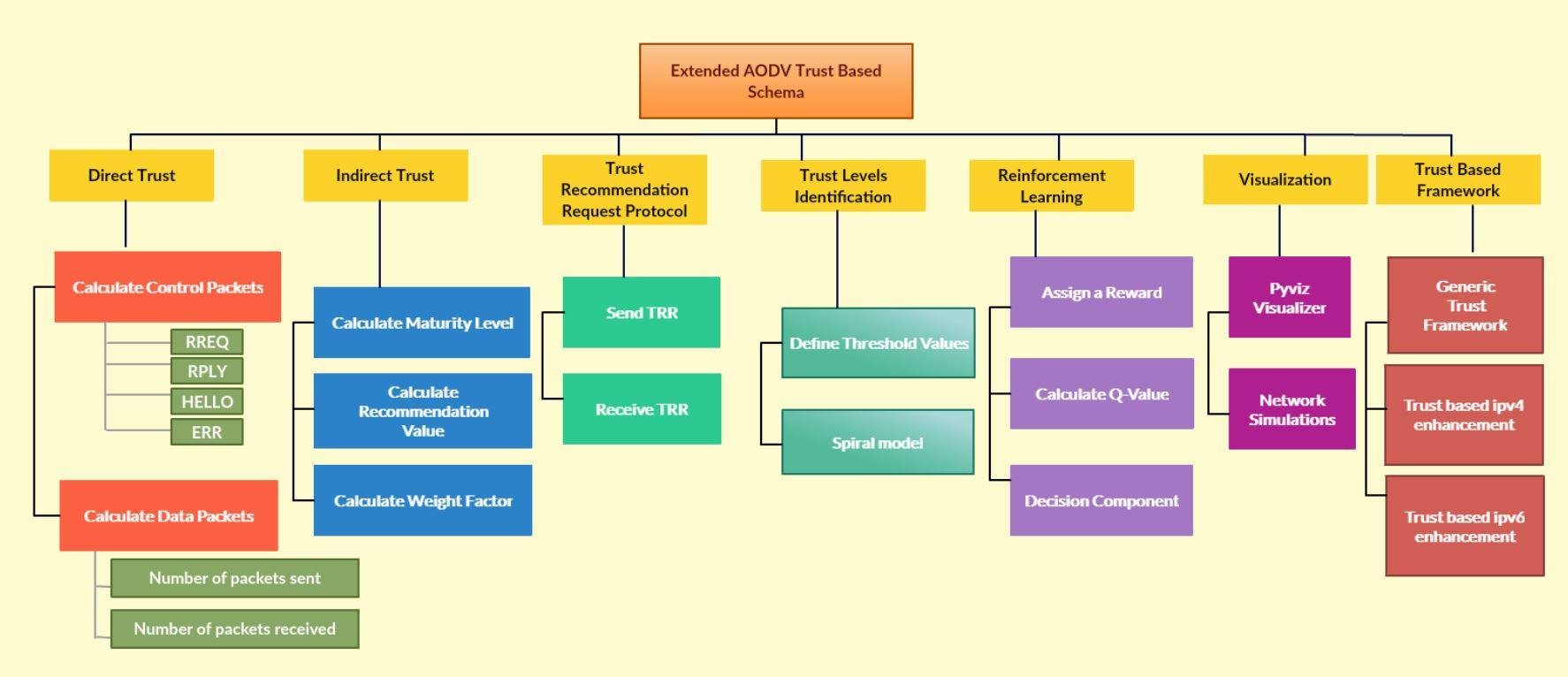
### **2.6.4. Generic platform in NS-3 for academic researchers**

As an initiative within the ns-3 community, we are developing a generic trust-based framework. This is really beneficial for the other researchers who are trying out to simulate the trust based protocols for other existing protocols in ns-3. In future, ns-3 users will find it really useful as a matter of little change in the simulation, Ipv4 and Ipv6 protocols can be turned into trust based framework and compare the improvement in many aspects.

# 3. Project plan

*Comment:  The time line that indicates the time frame within the various parts of the project to be completed.*

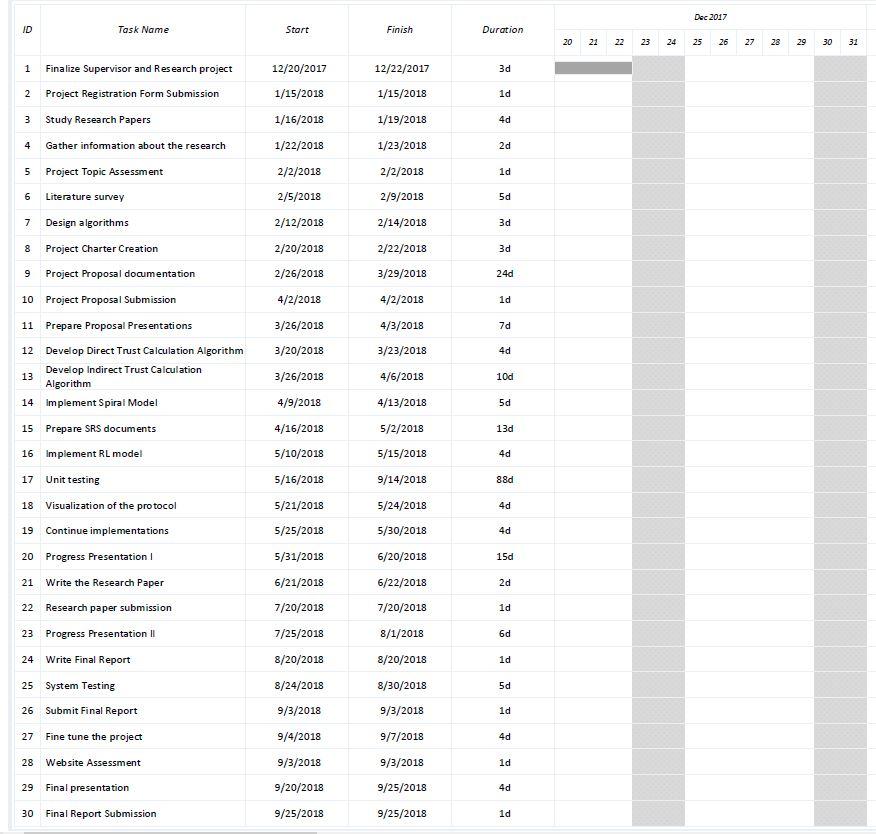
## 



**Figure [3.1]: Work breakdown structure**

**3.1. Workload Distribution**

|  |  |  |
| --- | --- | --- |
| **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. |



## Appendix [A]: Gantt Chart

# 4. Research constraints

## 

The suggested trust protocol is a general solution for MANET routing. In order to implement the protocol, researchers should be able to do simulations and collect necessary data. Even after the implementation is done, the product should be tested on networks under different circumstances and make sure it is working as expected. Due to the factors like each routing device’s memory capacity, computation power, energy etc it is not possible to do this process with actual networks with physical devices. Meaningful data cannot be collected since not only the node behaviors, but also the physical factors of devices also interfere with the network activities. The purpose of this research is to implement a trust protocol and the device issues are not something that can be considered as a main factor.

Since each and every one of these factors of devices can be controlled with a virtual simulator, as mentioned in 2.5, NS-3 is being used for this project. Although NS-3 is a powerful tool for a range of possible applications, there are still some limitations.

**Credibility and Validation**

Although network simulators are mostly similar to the actual network environment, all aspects of reality can never be implemented in a simulation. Regardless of how much the simulation designer has tried to make the simulator same as the real environment, it is never possible to get all the aspects right.

**Scalability Limitations**

One of the most important benefits of a network simulator is being able to add and remove devices anytime without complications. According to reference [12], the memory usage and the required computation time have been identified as two main limiting factors for scalability. Each node in the network requires memory space and therefore, the number of nodes which can be added will depend on the available memory. Also, in NS-3, there is a certain amount of time for events to be processed. Depending on the maximum amount of time the NS-3 designers have allocated for computation and simulation, the number of events that can be processed will be limited.

**Possible Malfunctions of Functionalities**

Simulation designers do not build all their functionalities from the scratch. They also make use of the existing models. Each one of these models are possible of either being failed or having wrong behavior at times. So it cannot be expected for all of the features to not have any malfunctions.

**5. Specified deliverables**

AODV, DSDV, and DSR are some of the most popular routing protocols for MANET which are used to established correct and efficient path between nodes. Among these, the trust-based framework which have been planned on implementing is for AODV is an on-demand routing protocol. This will be implemented using NS-3 network simulator where there’s a module for AODV protocol. If the finished trust-based framework is approved by the NS-3, it will be plugged in to the AODV module of NS-3 after the implementation.

As the main objective, this trust-based framework will make sure that nodes will select the most trustworthy route for forwarding packets instead of always selecting the shortest path. Therefore, the network will achieve benefits such as avoiding security attacks in a MANET such as blackhole attack and wormhole attack. If the end product is approved, this trust-based routing protocol will be deployed with NS-3 network simulator where other developers and users get to use this protocol for their network simulation purposes.

# 

# 6. 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] 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.

[12] Richard M. Fujimoto, Kalyan Perumalla, Alfred Park, Hao Wu, Mostafa H. Ammar, and George F. Riley. “Large-scale network simulation: how big? how fast”. *In In Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems (MASCOTS)*, 2003.