NS3 pseudo code documentation

1. Direct Trust

Trust that builds between two network nodes which are directly connected is called a direct trust. For given two network nodes, we can calculate the direct trust(DT) value by going through the following steps. When it comes to DT, control packets, and number of data packets which transferred between those two nodes are being considered. Number of RREQ , RPLY , ERR, HELLO packets are considered as control packets.

NDF (number of data packets sent) , NDR (number of data packets received)

CP (Control Packets)

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

DP = NDF / NDR

For a given node, it calculates the Direct Trust using CP and DP.

DT = CP + DP

For all nodes, we calculate the trust for their neighbour nodes and store those information in a table called **Trust Table**.

**Procedure :** Calculate Direct Trust

1. Identify neighbour nodes
2. for every neighbour\_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
8. get number of data packets sent
9. get number of data packets received
10. calculate DP
11. calculate DT
12. add DT to Trust table
13. end for

BEGIN

FOR each neighbour 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 DataPackets Sent

received = number of DataPackets Received

DP = sent data packets / received data packets

DT = CP + DP

Save DT in trust table DT to Trust table

END FOR

END

1. Identifying Trust levels

In this phase, network identifies the trust level of each node. There are five trust levels. ‘Global Trust’ value of each node is compared with the threshold values and the node’s trust level is decided accordingly. This ‘Global Trust’ is calculated by taking the average values of ‘Direct Trust’ and ‘Indirect Trust’ (GT = (DT + IT)/2).

Although the ‘Global Trust’ value is considered to make this decision, at the initial state we still do not have values for ‘Indirect Trust’ in the ‘Trust Table’. Therefore, initial state we consider only ‘Direct Trust’ instead of ‘Global Trust’.

*1. Trust Level 1 - Trustworthy Nodes (0.8 < TH)*

*2. Trust Level 2 - Partially Trustworthy Nodes (0.5 < TH <= 0.8 )*

*3. Trust Level 3 - Selfish Nodes (0.3 < TH <= 0.5)*

*4. Trust Level 4 - Pure Malicious Nodes (TH <= 0.3)*

*5. Trust Level 5 - Collaborative Malicious Nodes (TH <= 0.3)*

**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 > 0.5, **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 > 0.3, **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**

Below is the pseudo code we came up with for this procedure.

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> 0.5

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>0.3

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

1. Spiral Model

Collaborative malicious node discovery process(Spiral Model) is used to distinguish pure malicious nodes and collaborative malicious nodes(CM). Basically it compares the current trust value against its past trust values. When calculating past trust values we will retrieve the highest trust value and lowest trust value by analyzing its past records. If current trust value is not an outlier, we assume it’s a pure malicious node because it’s not showing an abnormal trust value. But if the current trust value is an outliers we can suspect it as a collaborative malicious node. So we can basically consider it may be the collaborative malicious node which may be connected to some other malicious nodes.

**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 pure malicious node.

4: Then it(the node who execute this) can delete that record from all of it's 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 it's 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.

Below is the pseudocode we came up with.

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

1. After Transmission Phase

In this phase, network is reacting to already identified collaborative malicious nodes in the network. Once a node has broadcasted a message, the receiving node will act upon that message by running this particular algorithm. Basically this after transmission phase is being executed once a broadcast message is received from any node in the network.

This includes, check whether the received collaborative malicious node has already blacklisted, then we simply remove it from all trust tables. Because it has been confirmed for the 2nd time by some arbitrary node in the network.

If the received node is not previously marked as blacklisted, then mark it as blacklisted in all trust tables.Then we broadcast the message to the other neighbour nodes.

Finally, check if this particular collaborative malicious node has been recommended by some other node. If so, we recalculate their trust values as well. Because this was an incorrect recommendation.

CM = Collaborative Malicious

**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

//B received broadcast messages from A that C and E are CM nodes. So B will execute this.

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 neighbour 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