Circuits and Systems, 2016, 7, 3228-3245
Published Online August 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.710275

Scientific Research Publishing

A Compact Trust Computation and Management Approach for Defending against Derailed Attacks for Wireless Sensor Networks and Its Applications

R. Mohan Kumar, A. V. Ram Prasad

Department of ECE, K.L.N. College of Engineering, Sivagangai District, India
Email: psrmohan2003@gmail.com

Received 10 May 2016; accepted 30 May 2016; published 24 August 2016

Abstract

One of the most effective measurements of intercommunication and collaboration in wireless sensor networks which leads to provide security is Trust Management. Most popular decision making systems used to collaborate with a stranger are tackled by two different existing trust management systems: one is a policy-based approach which verifies the decision built on logical properties and functionalities; the other approach is reputation-based approach which verifies the decision built on physical properties and functionalities of WSN. Proofless authorization, unavailability, vagueness and more complexity cause decreased detection rate and spoil the efficacy of the WSN in existing approaches. Some of the integrated approaches are utilized to improve the significance of the trust management strategies. In this paper, a Compact Trust Computation and Management (CTCM) approach is proposed to overcome the limitations of the existing approaches, also it provides a strong objective security with the calculability and the available security implications. Finally, the CTCM approach incorporates the optimum trust score for logical and physical investigation of the network resources. The simulation based experiment results show that the CTCM compact trust computation and management approach can provide an efficient defending mechanism against derailing attacks in WSN.

Keywords

Wireless Sensor Networks, Trust Management, Security, Intrusion Detection System, Malicious Attacks

How to cite this paper: Kumar, R.M. and Prasad, A.V.R. (2016) A Compact Trust Computation and Management Approach for Defending against Derailed Attacks for Wireless Sensor Networks and Its Applications. *Circuits and Systems*, **7**, 2328-2345. http://dx.doi.org/10.4236/cs.2016.710275

1. Introduction

Vulnerability is a shortcoming in a framework that can be abused to contrary that affects classification, respectability, and/or accessibility. There are numerous ways in which vulnerabilities can be classified. This article discusses about three abnormal state defects classes: programming defects, security design issues, and programming highlight abuse. These classes are explained below.

A product defect is brought on by an unintended mistake in the outline or coding of programming. An illustration is an information approval mistake, for example, client gave information not being legitimately assessed for malignant character strings and excessively long values connected with known assaults. Another illustration is a race condition mistake that permits the assailant to perform a particular activity with raised benefits. A security design setting is a component of a product's security that can be modified through the product itself. Case of settings having some working frameworks that offer an access to the control records benefits for clients, and an application setting is offered to handicap with encryption of information which puts away by the application. Security setup issue vulnerability includes the utilization of security arrangement settings that adversely influence the security of the product. A product highlight is a utilitarian ability given by programming. Product highlight abuse helplessness is a defenselessness in which the element gives a road to bargain the security of a framework [1]. These vulnerabilities are brought on by the product architect making trust presumptions that allow the product to give gainful components, while additionally presenting the likelihood of somebody who is abusing the trust suppositions to trade off security. For instance, email customer programming may contain a component that renders HTML content in email messages. An aggressor could create a deceitful email message that contains hyperlinks that, when rendered in HTML, appear to the beneficiary to be kindhearted. All things considered take the beneficiary to a malevolent site when they are tapped on. One of the trust suppositions in the outline of the HTML content rendering highlight was that clients would not get malignant hyperlinks and click on them [2].

Vulnerability can be deployed automatically or manually by malicious people in terms of hardware and software or through any kind of network activities like entering into a network without permission, injecting defects into data packets, intruding data packets for misuse, transferring a data packet in a wrong route and fetching an un-authorized data from the storage. All kinds of malicious activities are also known as vulnerabilities. Vulnerability spoils the network behavior and reduces the customer satisfaction. Also the application developed under WSN is deployed and used for remote, emerging and advanced industries like military, surveillance and healthcare.

A trust management scheme can be used to aid an automated decision-making process for an access control policy. Since unintentional temporary errors are possible, the trust management solution must provide a redemption scheme to allow nodes to recover trust. However, if a malicious node tries to disguise its malicious behaviors as unintentional temporary errors, the malicious node may be given more opportunities to attack the system by disturbing the redemption scheme. Existing trust management schemes [3] that employ redemption schemes fail to discriminate between temporary errors and disguised malicious behaviors in which the attacker cleverly behaves well and bad alternatively. This character is called as trust whereas the performance is calculated by the likelihood ratio for given activity. A trust administration plan deals with the trust by incorporating the ideas of qualifications, access control, security arrangement, accessibility, and verification. By utilizing the incorporated data, a trust administration plan can be utilized to help a computerized basic leadership process for an entrance control arrangement.

Trust can be assessed in an assortment of ways. Direct perception assesses neighboring nodes by watching their conduct. For instance, in a WSN, a node can identify malignant neighbors by checking what numbers of packets were sent to the following node [4]. Additionally, if a source node analyzes the substance of the bundles, it can identify manufacture alternately alteration [5]. With backhanded perception nodes distribute their immediate perceptions to their neighboring nodes to caution about malevolent nodes or to report recuperated nodes that were already assessed as malignant. In a WSN, a cautioning message from different nodes will prohibit the vindictive node from the system. Then again, recuperation reports from different nodes can permit nodes to rejoin the system [6]-[9]. There exist trust administration plots that address the proposed assault in different types of systems, for example, Traditional Networks [10], Cognitive Radio Networks [11], Peer-to-Peer Networks [12], Ad-hoc Networks [13], and Wireless Sensor Networks [14] [15]. Be that as it may be, no trust administration plan can separate between interim blunders and on-off assaults. In the above discussed initiator

proposed that the neighboring hubs should be assessed by utilizing immediate and roundabout trust values. The arrangement depicted in one of existing studies, utilizes a gathering of immediate and roundabout assessments to lessen the trust of an on-off aggressor to be lower than the trust of other neighboring typical hubs, so the system will reroute around the on-off assault hubs in the framework.

2. Trust Calculation and Trust Management

To start with, every node watches and stores the neighboring nodes' practices as indicated by the area esteem they gave. Second, every node gathers and stores the notices or reports from different nodes about its neighboring nodes. Third, every node Figs the trust in light of the conduct data gathered and put away for each neighboring node. Keep going, in light of the trust and the approaches that utilization the trust, every node chooses the best node or gathering of nodes with which to team up. Those teamed up best nodes are utilized to make a most limited way to transmit information bundles. In this paper, a compact trust management system is proposed to transmit data in a trusted manner by investigating nodes and selected trusted, energy efficient nodes. The whole usefulness of the CTCM way is to deal with to get a best throughput is portrayed in the accompanying areas. Trust management schemes aim to improve collaboration between the entities in a distributed system by predicting future behaviors of peers based on their previous behaviors. A trust management scheme typically does this using the following steps.

- Node behavior analysis by ID.
- Node location.
- Node mobility speed.
- Data packet format used for transmission.

2.1. Network Assumptions

Here, it is assumed that the network taken to simulate the proposed approach is denoted as G, is a collection of vertices *V* and edges *E*.

$$V = \left\{node_1, node_2, \cdots, node_N\right\} \tag{1}$$

$$E = L_{ij} = Link\left(node_i, node_j\right) \ \forall i, j \in V \tag{2}$$

Some of the attributes and their values are assumed to construct the network is given here and it is followed from [16].

a) Number of Nodes available in the network G is N
b) Location index for each node-iis denoted as (x_i, y_i)
c) Each node-i is assigned with ITV is 100. // Trust value
d) Number of paths possible from S1 to D1 is M
e) Number of Shortest paths can be considered from S1 to D1 is K
f) Raw Distance from S1 to D1 is Dis1
g) Next hops or neighbors between S1 and D1 is $\left\{h_1, h_2, \cdots, h_m\right\}$
h) Distanced obtained from path K, for S1 to D1 is OptDis.
i) Network G, the area of G is R x W. Where R is the width of the G and W is the Height of the G.
j) Also, it is assumed that the network size is scalable to the CTCM approach.

Source node S1 and destination node D1 is selected from network G, where each node is having energy of 100 Joules initially. For every action each node spends some amount of energy such as transmission, receiving, listens and idles, etc. After each activity the initial energy is reduced.

Where N nodes are located in F-dimension, and divided into level by level from source node to destination node which is shown in **Figure 1**. All the nodes in the network are connected in a tree form in level by level (like parent child). The number of nodes in each level is k where the number of level in the tree is p. So that, the total number of intermediate nodes between S1 to D1 can be represented as:

$$\frac{N}{p^k} - 1\frac{\mathrm{N}}{\mathrm{p^k}} - 1 \tag{3}$$

The tree construction needs many multiplications and additions

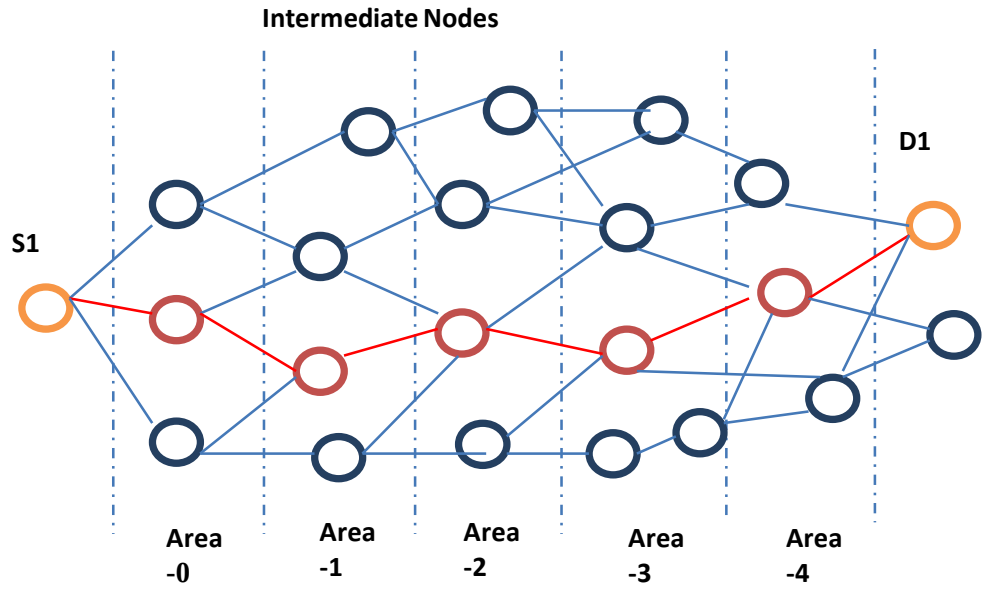

Figure 1. Neighbor selection for route discovery.

$$\sum_{k=0}^{(\log N)-1} p^k \cdot F \cdot \left(\frac{N}{p^k} + Q \right) \tag{4}$$

$$\sum_{k=0}^{(\log N)-1} p^k \cdot F \cdot \left(2 \cdot \frac{N}{p^k} - 1 \right) \tag{5}$$

After constructing the network as a tree, it is easy to verify the intermediate neighbors in a sequential order, the idea behind is, it should not miss any node in the network. This method improves accuracy of the neighbor selection within N number of nodes. After successful selection of the current energy of the neighbor node is computed, to choose as optimal neighbor.

2.2. Node Location Verification

During path discovery the intermediate node gives location identification; monitor node in the network receives data from the source node to base station. BS is a high energy node acts as a sink. Grid has node location which identifies for node placement. Source-Destination (SD) node sends the beacon message packet to the adjacent nodes in the path. The sensor node which receives the message packet updates the message format and sends back to SD node. SD node format is specified in **Figure 2**.

The header format of SD node consists of Zone Id, in which zone the sensor node belongs. Source node id field represents which node is sending this header information packet. Say if SD node is sending to the sensor node then source node id is SD node ID. Destination node ID specifies which node is to respond the message packet. Next field represents Timer; timer is an atomic clock which runs in SD nodes. It appends the sending time in milliseconds to the message format. According to the node and node type it is assessed that it is a normal sensor node or monitoring node. Synchronization flag is to set when corresponding destination node is tried to communicate the message packet. Next field is used to append message to ensure the proper communication between sensor and SD nodes. Last field indicates end of message format header. Example for this is specified in **Figure 3**. SD node called G6 is to identify or estimate the zonal nodes which is neighbor to it is framed in the message format. In **Figure 4**, the first field "3" represents node actions are carried out in third zone. SD node called G6 is a GPS unit node used in this zone to estimate the other sensor node's locations. Once adjacent node is identified by this SD node it frames the message packet by filling the destination node ID, sensor node ID, and clock time is appended in the message packet, and synchronous flag is set to True, and corresponding message is appended along the packet with delimiter as EOF. Here the message packet SD node G6 is CTCM to

Area ID	Src Node ID	Dest Node ID	Timer in (ms)	S/M/G	Sync Flg	Msg	EOF

Figure 2. Message format.

3	G6	12	3200ms	S	True	Msg	1

Figure 3. Message format by SD Node G6.

3	12	G6	3200ms	G	True	Msg	1

Figure 4. Message format by sensor node 12.

send the packet to the sensor node 12 to identify the message packet status and to estimate the node location. Say 3200 ms is the start time which is taken into consideration to send the message packet and appended in the header format.

The way BS connected with the other sensor nodes in the network is illustrated in **Figure 5**. In receiving node the header format gets updated without replacing timer and it send back to STD node. Extract the timer information from the reception packet STD node calculates the time delay as follows.

$$\text{Time delay} = \text{current atomic timer} - \text{extracted timer} \tag{6}$$

with this time delay SD node can easily estimate the location of the adjacent node.

2.3. Location Estimation

The node location, distance among the nodes calculated (see **Figure 6**) for location based trust verification leads to construct a trusted path from source to destination. The probability ratio of trusted path calculation is shown in **Figure 7**.

$$\langle \text{Time}, \text{direction}, \text{stats} \rangle \Rightarrow \text{Node location Weight matrix}$$

$$\Rightarrow \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(D_{ij} - d_{ij} \right)^2 \tag{7}$$

$$d_{ij} = \sum_{k} x_{ij}, kU_k \tag{8}$$

$$Q = \sum_{i=1}^{n} \sum_{j \neq 1}^{n} w_{ij} \left(D_{ij} - \sum_{k} x_{ij}, kU_k \right)^2 \tag{9}$$

$$X^{\mathrm{T}} D = \left(X^{\mathrm{T}} X \right) \cdot V \tag{10}$$

Least square length

$$V = \left(X^{\mathrm{T}} X \right)^{-1} \cdot X^{\mathrm{T}} D \tag{11}$$

Sum of the squares will be

$$Q = \sum_{i=1}^{n} \sum_{j,j \neq 1}^{n} \frac{\left[D_{ij} - ED_{ij} \right]^2}{\operatorname{var} D_{ij}} \tag{12}$$

Based on the distance and time, STD node sends and estimates the zonal sensor node. Length of the node identity differs with the other nodes. For the estimation, number of nodes which is considered for the node discovery is explained in **Figure 7**. The probability of accessing the nodes is depicted.

The STD node works in the distance matrix algorithm. Group distance is obtained for two positive integers, and the algorithm is specified below,

Algorithm-1: Distance_Matrix ( )

{

1). Find m and n which have minimum distance of D_{mn}.

2). Assume the group (m n) which constitutes $n(mn) = n_m + n_n$ members.

3). Based on the direction and time correct m and n to the STD node to form a group (mn).

(*i.e.*) two paths connecting m to (mn) and n to (nm)

$$\text{length} = \frac{D_{mn}}{2} \tag{13}$$

4). Distance of this group obtained is computed except for m and n by,

$$D_{mn},\gamma = \left(\frac{n_m}{n_m + n_n}\right) \times D_{m\gamma} + \left(\frac{n_n}{n_m + n_n}\right) D_{n\gamma} \tag{14}$$

5). Remove the matrix row and column relates to m and n and add matrix properties for (mn) group.

6). Continue algorithm if data matrix has one value else return to step (1).

}

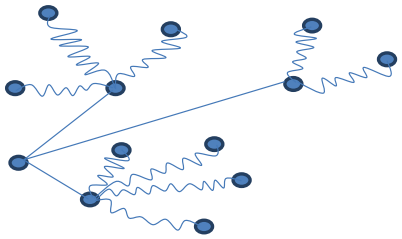

Figure 5. BS Node interaction with SD node.

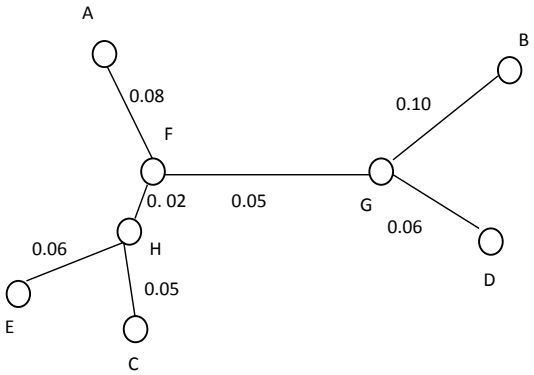

Figure 6. Location estimation and distance calculation.

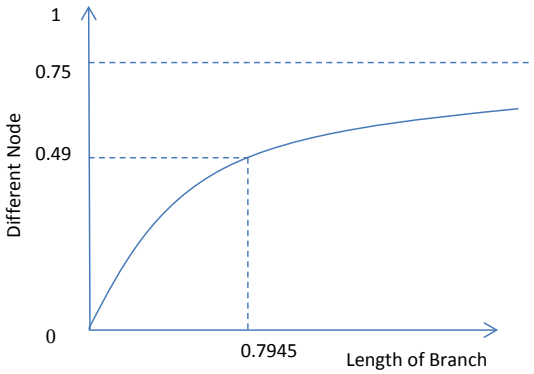

Figure 7. SD Node path discovery.

The trusted path can be constructed using the following **Figure 8** where only the trusted nodes are interconnected together to construct the trusted path.

Groups

if $\{(N_1, N_2) \Rightarrow A$,

$(N_2, N_6) \Rightarrow B$,

$(A, B) \Rightarrow C_1$,

$(N_1, C_1) \Rightarrow D_1$, $D_1 \Rightarrow \varepsilon$

else

$\emptyset\}$

The connectivity which is obtained with sensor node and STD node is grouped based on the existence of node. It is possible to group the sensor node and then mapping it to the STD node. The complete rule of grouping mechanism is depicted in **Figure 8**.

Distance calculated from STD node to direction node is,

$$V_m = \frac{1}{2} D_{mn} + \frac{1}{2}(V_m - V_n) \tag{15}$$

$$V_n = \frac{1}{2} D_{mn} + \frac{1}{2}(V_n - V_m) \tag{16}$$

All sensor nodes is calculated by

$$D(m,n)\gamma = \frac{D_{m\gamma} + D_{n\gamma} + D_{mn}}{2} \tag{17}$$

$$Node \Rightarrow (x_i, y_j, z) loc \tag{18}$$

$$direction \Rightarrow \frac{0,\ 1,\ 2,\ 3,\ 4(node)}{\uparrow,\ \downarrow,\ \leftarrow,\ \rightarrow,\ \updownarrow} \tag{19}$$

is obtained by distance matrix calculation

The distance matrix calculation and its direction are calculated for the simulation purpose, GPS strength is monitored with the cost value in the zone. Benefit value is assumed 350 and above. The strength of GPS lies in medium to high range. It is depicted in **Figure 9**. The positive rate and Strength of GPS node is compared and its probability lies in $0.5 < x < 1$. The probability ratio of actual location value comparing with the predicted location value is given in **Table 1**.

GPS node called STD node acquires the properties of other sensor nodes in the zone, based on the distance metric, each sensor node distance is identified and normalized by weight matrix is shown in **Figure 10**.

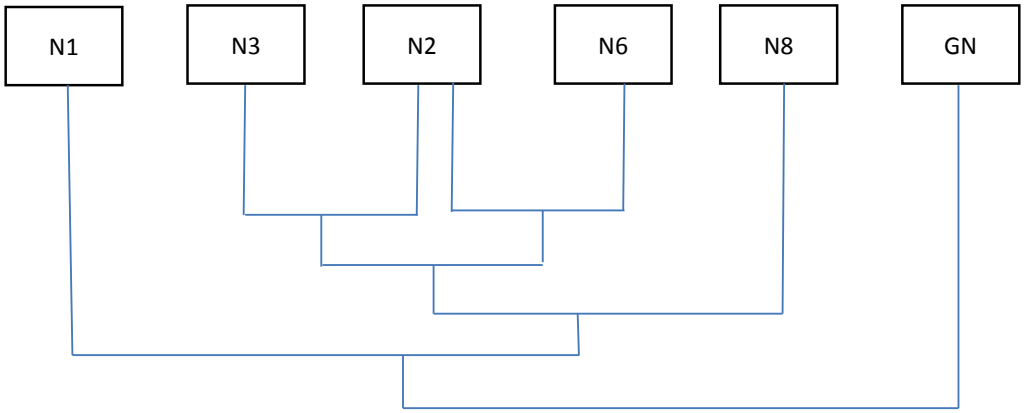

Figure 8. Grouping of STD node with sensor node.

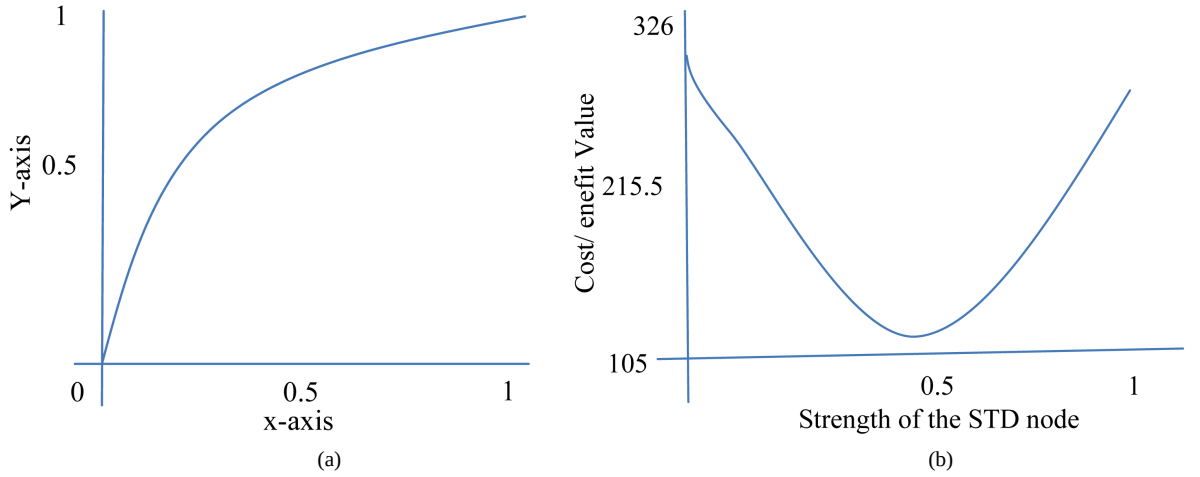

Figure 9. (a) Comparison of strength of GPS node and positive rate; (b) Strength of GPS node and cost benefit value.

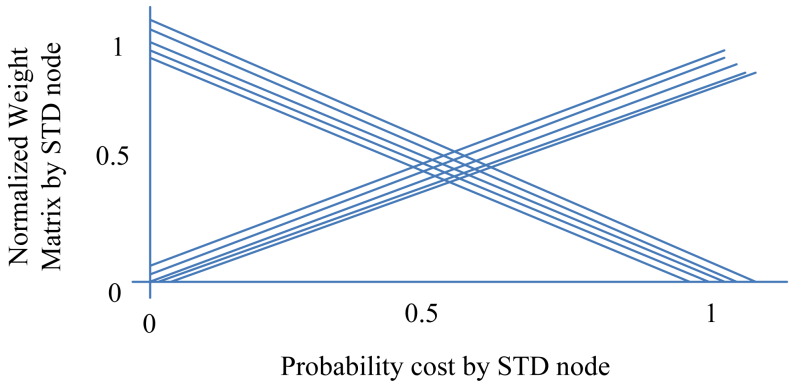

Figure 10. Weight matrix normalization.

Table 1. Location comparison.

Type	Probability	Value
Predicted	45.67%	274
Actual	54.33%	326

For simulation, numbers of nodes are restricted to maximum 75. There are four zones along with STD node dispersed in the zone. Success rate among the nodes are under the probability factor. When number of node increases the STD node also increases and placed in the centric approach which can access all the sensor and monitor nodes in the network.

GPS Node: Location Identification

MN: Rx data from source to Tx

BS: High energy node acts as a sink

Extract clock time from GPS node, subtract from sent time, hence the current service time is identified. If it falls, GPS node has information of sensor nodes (**Figure 11**).

$\left(X_{ap}, Y_{ap}\right)$ ⇒ Actual position of sensor nodes

$\left(X_{est}, Y_{est}\right)$ ⇒ Estimated Position of sensor nodes

Predicted Error

Average Error

Wmax ⇒ Wt

1) Adjacent nodes and connectivity of neighbors is identified
2) STD node and its energy is computed
3) Wt is calculated

4) $\left(X_{est}, Y_{est}\right) = \sum_{i=1}^{k}\left[\frac{w_i X_i}{w_i}, \frac{w_i Y_i}{w_i}\right]$ (20)

Start → adjacent SD nodes → weight matrix → Location Estimation → stop

Rules

1. If obtained delay time is very low than P = [ ]
2. if it is low, medium, high, very high
3. Predicted Error = $\sqrt{\left(X_{est} - X\right)^2 + \left(Y_{est} - Y\right)^2}$
4. Average PE = $\frac{\sum(A)}{N}$, N ⇒ Number of sensor nodes.

According to the number of nodes deployed in the network, and number of rounds applied, selecting the STD nodes in terms of node exploration is shown in **Figure 12**. The number of STD nodes is positively proportional to the number of nodes deployed in the network. The entire process of the proposed trust management is given in the form of algorithm, which is given below (Algorithm 2). Similarly, the step by step procedure followed in the proposed approach is illustrated in **Figure 13**. Nodes are verified by tracking their current information in the network. According to the present location, distance and the trust value the node is decided as a trusted node or not. The probability ratio of number of node tracking against the number of nodes deployed in the network is shown in **Figure 14**.

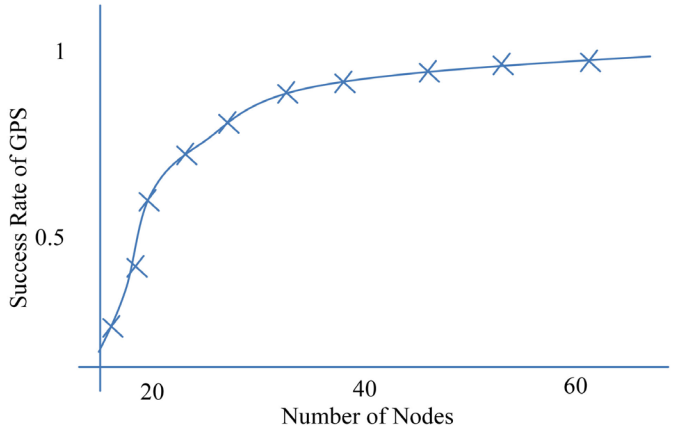

Figure 11. GPS success rate in network zone.

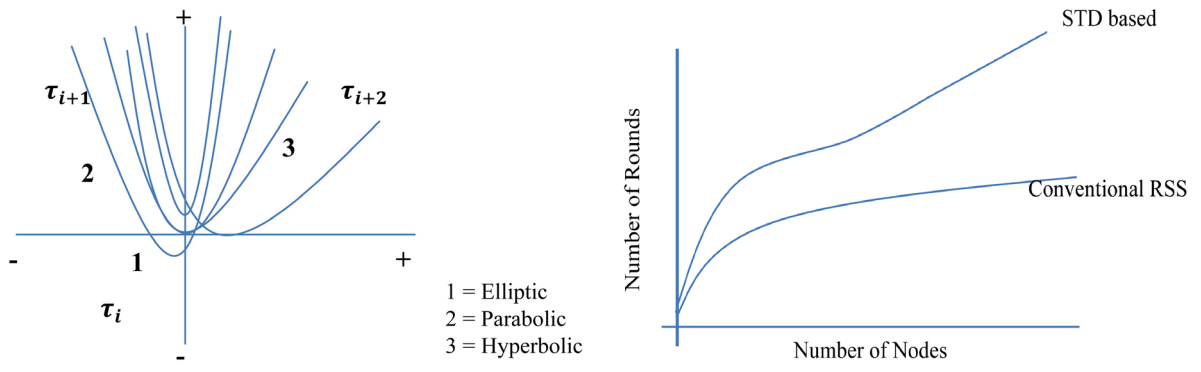

Figure 12. Node exploration by STD node.

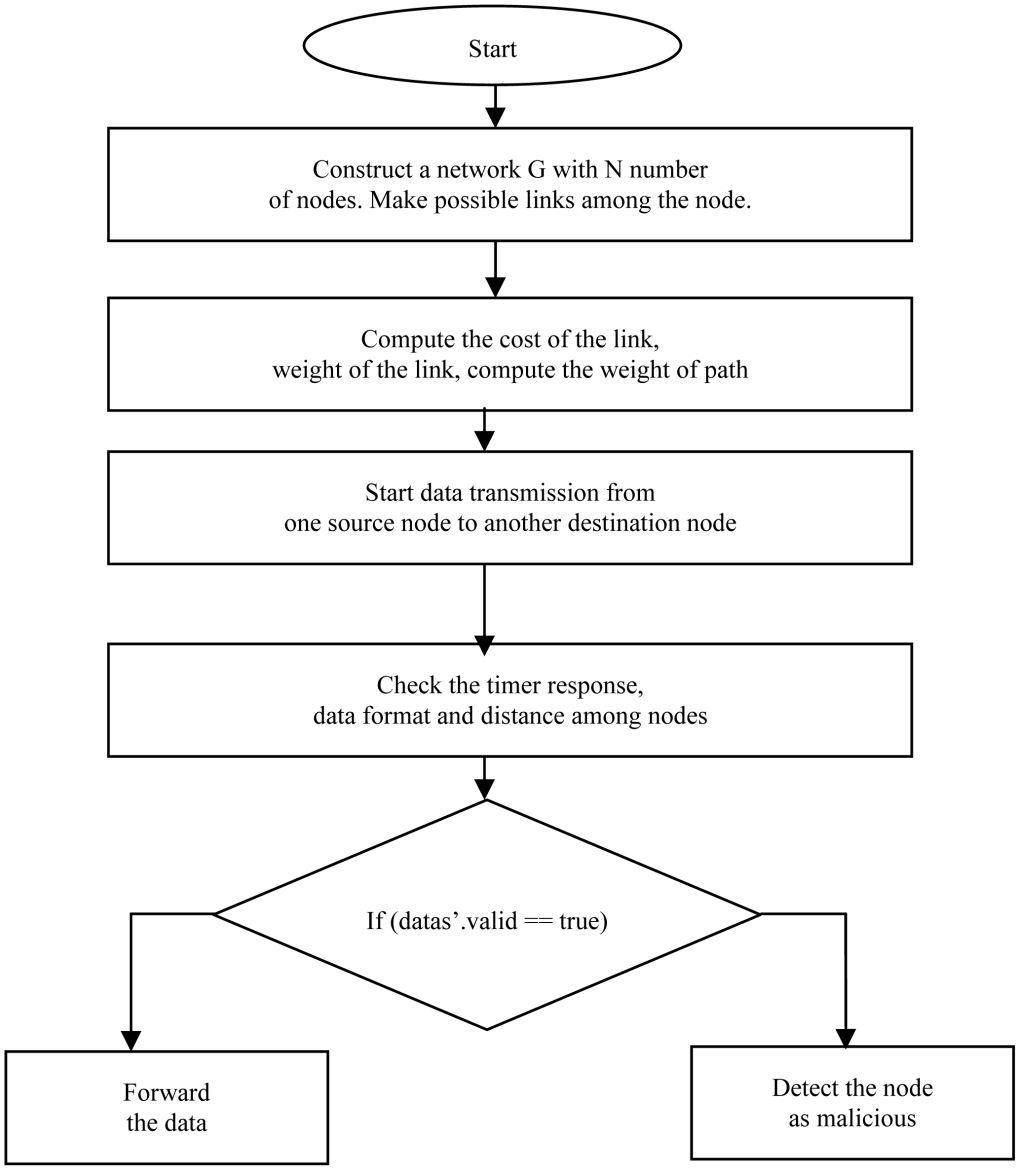

Figure 13. Flowchart for proposed approach.

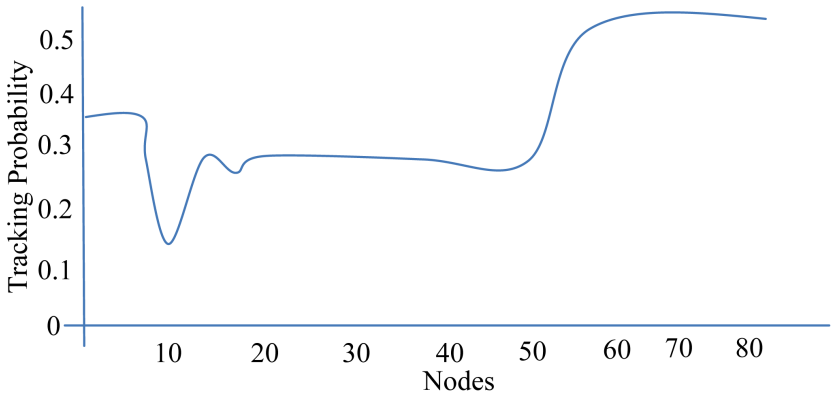

Figure 14. Node vs. tracking probability.

Algorithm-2:

1. Graph network (N, E) $\Rightarrow$ G
2. N = Number of nodes
3. (N, STD_N , M_N) $\in$ G(N)
4. E = Link set
5. C(X, X') = cost of link (X, X')
6. W(X, X') = weight of link (W, W')
7. cost = $\begin{cases} 1, \forall all\ i \\ \frac{1}{BW}, Inverse\ BW \\ \frac{1}{cong}, inverse\ congestion \end{cases}$
8. weight of the path, $(X_1, X_2, \cdots, X_n) = W(X_1, X_2) + \cdots + (X_{m-1}, X_m)$
9. $N' = \{STD_N\}$
10. for all nodes in N' then
11. if "v" next to "u" then
12. frame-header(Zid, Sid, Did, clck-time, node-type, time, msg, 1)
13. dist(v) = c(u, v)
14. extract timer response
15. call distance matrix algo
16. if (timer-response) is within the range1 then
17. dist(v) = estimate (STD_N , v, thd_1)
18. else if (timer-response) is within the range2 then
19. dist(v) = estimate (STD_N , v, thd_2)
20. else
21. dist(v) = out of range / fails
22. end if
23. dist(v) = min (dist(v), dist(w), cost(w, k))
24. call distance algorithm
25. clock-time = extracted timer- actual timer
26. if (dist(θ) $\approx$ 0) then
27. the given node is in parabolic range of STD_N
28. if (dist(v) < 0) then
29. the given node is in elliptic range of STD_N
30. else
31. the given node is in hyperbolic range of STD_N
32. end if
33. end if
34. calculate for all odes in the zone
35. end loop
36. end algorithm

Theorem 1

Atomic timer $\{SD_{Ni}, \cdots, SD_{Nj}\}$ is a set of timer based sensor tracking for detection nodes, which has internal counter for increments the clock-timer, appends to the master header format for node exploration in time

$$\{\tau_i, \tau_{i+1}, \cdots\} \tag{21}$$

$$SD_{Ni} \Rightarrow \{N_1, N_2, \cdots, N_i\} \text{ and } Z_{id} \in SD_{Ni} \tag{22}$$

$$SD_{Nj} \Rightarrow \{N_1, N_2, \cdots, N_j\} \text{ and } Z_{id} \in SD_{Nj} \tag{23}$$

Theorem 2

Timer is a counter clock appended in SD node which has a high resolution and reset the every successive estimation done in zone $\{Z_1, \cdots, Z_n\}$.

SD_{Ni} (Timer) is alive when

for all nodes in $Zone_i$

dist (N_1 , SD_{Ni}). estimate(flg) is true

end for

if estimate(flg) is false

reset timer of SD_{Ni}

end if

end

Theorem 3

Let G is a network and Z' is the zone in the network G, which is $\{Z_i', Z_j', \cdots, Z_m', 1\} \in G$ such that every node Ni in zone set called $\{Z_k'\}$, $|N_i \cup Z_k'| \geq G'$, where G' is a sub network.

Proof:

for all j nodes,

$N_j \geq 0$, the graph network $\propto_{\beta}^{j} Z$ is a part of G and G'.

$\propto \cdot Z' \neq \beta Z'$ (or) $|\propto| > 0$ (or) $|\beta| \geq 1$

every node($N_i \in Z_k^i \in \& / 1$ and it's a part of G and G' in Z_i'.

3. Trust Value Based Node Selection

The trust estimation of every node is confirmed amid the course revelation and information, Transmission in a course is to enhance the secured transmission. In this work every node is checking its neighbor node's conduct and relegates a Trust Value (TV) to give trustiness. The trust worth is spoken to by a couple of quality is {−1, 1} where −1 indicates the negative conduct and 1 signifies the positive conduct of the neighbor node. At whatever point a node comes into the system, its essential and fundamental information is confirmed and, the ITV worth will be given by the system administration. The ITV is introduced at 100 and the estimation of ITV gets fluctuated for each course disclosure. In order to compute the trust value for every time, direction and the state of the node is computed. A sample computed data is given in **Table 2**. In this table, the node with the time, direction and the node status whether node is closer to the STD node or not etc., is given clearly

Amid the season of observing, there are two qualities can be conveyed as nID-[Neighbor ID] and nBH-[Neighbor Behavior]. In the event that the nodes' ACK-time, RPLY-time PKT-size, Direction are relegated with an alternate worth than the ordinary esteem then the $nBH = \varnothing$ else $nBH = 1$. The checking capacity is

$$fm(h_1, h_2) \text{ returns a triple } \langle nID, nBH, nLc \rangle \tag{24}$$

where, the value of the tuple is

$$\begin{cases} \langle h_1, 0 \rangle & \text{if } nLC = 0 \left[\text{location is accurate}\right] \\ \langle h_1, 1 \rangle & \text{if } nBH = 1 \left[\text{behavior is good}\right] \\ \langle h_1, -1 \rangle & \text{if } nBH = -1 \left[\text{behavior is bad}\right] \end{cases} \tag{25}$$

Table 2. Time calculation.

Time (ms)	Direction (0/1/2/3)	Stats (0/1)	Node of SD_N
96(N4)	0	1	Node is very near and it is in left to STD
126(N6)	1	1	Node is in range of τ_{i+} right side
42(N9)	2	1	Node is adjacent to STD_N and it's above.
64.52(N14)	3	1	Node is below to STD_N and it's below
792(N17)	NA	1	Node beyond limit

According to the tuple value given by the monitoring node, the *ITV* is decreased when $nBH = -1$, else is not. The Node can be accepted in the route discovery, whenever the $ITV > \delta_{TV}$. Where δ_{TV} is the threshold value of *TV* of the node. By utilizing the trust value given by the neighbor node and the trust value assigned by the management are combined together and a trust score is computed for node's trust evaluation.

3.1. Shortest Path Selection

Since node S1 and node D1 are located far from each other, it is necessary to transmit the data through some intermediate nodes (neighbors). The number of neighbors between the sources to destination depends on the size of the network and they are denoted as h_i. Path-I: that is from S1 to D1 constructed through J number of intermediate nodes, where each intermediate node is selected as the best neighbor using CTCM algorithm. Each neighbor nodes are selected under constraints as:

$$\begin{cases} dist(h_i) < dt \\ CEh_i < te \text{ and } CE(h_i) > h_x \\ TV(h_i) > tt \end{cases} \tag{26}$$

From (1), the neighbor node should be located within a threshold distance from the previous neighbor node, and it should have more current energy than the other nodes and it should have more energy than the threshold energy. Also, the TV of the neighbor node should be greater than the threshold trust value.

$$P(i) = dist(S_1, h_1) + dist(h_1, h_2) + \cdots + dist(h_i - 1, h_i) + \cdots + dist(h_m, D_1) \tag{27}$$

$$\text{Distance of the path } P(i) = \sum_{i=1}^{m} (h_i, h_i + 1) \tag{27a}$$

where

$$dist(h_1, h_2) = \sqrt{(h_{2x} - h_{1x})^2 + (h_{2y} - h_{1y})^2} \tag{28}$$

(h_{1x}, h_{1y}) is the (x, y) location coordinates of the neighbor h1 and (h_{2x}, h_{2y}) is the (x, y) location coordinates of the neighbor h2. Similarly, each path distance is computed and compared each other. Finally the minimum distance is obtained as the shortest path between the source node and the destination node and it can be written as:

$$\text{if } \left(dist(P_i) < \left\{dist(P_1), dist(P_2), dist(P_{i-1}), dist(P_{i+1}), \cdots, dist(P_m)\right\}\right) \tag{29}$$

then P_i, is selected as the shortest path between S_1 to D_1. After choosing the shortest path, the data can be transmitted from S_1 to D_1. The route construction can be obtained by providing a link between the pair of next neighbors, if it satisfies the following constraints:

$$Link_{node_i, node_j} = \begin{cases} Link(node_i, node_j) = 1 & \text{if } TV(node_i, node_j) \text{ and} \\ & CE(node_i, node_j) \text{ and} \\ & Dist(node_i, node_j) < threshold \\ 0, & \text{if } TV(node_i, node_j) \text{ and} \\ & CE(node_i, node_j) \text{ and} \\ & Dist(node_i, node_j) < threshold \end{cases} \tag{30}$$

$Link_{node_i, node_j}$, the process is repeated from source node to destination node for constructing a better path. $Link_{node_i, node_j}$, is divided into 3 ways where one is for the energy link ($LinkE_{node_i, node_j}$), second is Trust based link($Link_{node_i, node_j}$), and the third is distance based link Equations (7), (8) and (12) decide whether the node is valid and it can have a link or not. The values of the conditional parameters cannot have a constant value all the time and they can have a threshold value for computing its validity. It can be written as:

$$\sum_{i=1}^{N} CE\left(Node_i\right) \geq E_{th} \tag{31}$$

$$\sum_{i=1}^{N} TV\left(Node_i\right) \geq TV_{th} \rightarrow \sum_{i=1}^{N} Link\left(Node_i, Node_j\right) == true \tag{32}$$

From (14), it is clear that if the current energy of the node should be higher than the energy threshold value and the trust value of the node is also greater than the trust threshold value then the link between two nodes is possible else, search for the other neighbor node. Since the number of neighbor nodes is dynamic and the number of parameters is getting varied each time, the constraint based choosing the best neighbor node can be obtained using AIS algorithm.

3.2. Energy Selection

Energy selection is obtained by computing the consumed energy for data transmission, receiving, idle mode and for sensing the data. The current energy of a node can be calculated by subtracting all the consumed energy from the initial energy and it is written as:

$$CE\left(Node_i\right) = IE - \left[E_{Tx}^{kb} + E_{Rx}^{kb} + E_{Idle} + E_s\right] \tag{33}$$

where,

$CE(Node_i)$: Current Energy
IE: Initial Energy
E_{TX}(KB): Transmission energy for KB
KB: size of the data
E_s: Sensing energy
E_{Idle}: Idle energy
For each action of the network nodes, the amount of energy needed is assigned here.
Initial Energy = 100 joules
Transmitting Energy = 0.26 joules
Receiving Energy = 0.08 Joules
Idle Energy = 0.01 Joules
Sleep Energy = 0.005 Joules
Wakeup Energy = 0.005 Joules

Finally the trust value is assessed by computing the energy value, location based trust value, neighbor node assessment based trust value and other node behavioral based trust value (REQ-RES time) are combined to decide a node that is trusted node or not. This CTCM approach based trust value helps to detect and prevent Sybil, sinkhole, Selective Forward (SF) and on-off attack. Since after verifying the nodes' internal and external behavior all Sybil and sinkhole attacks are detected and eliminated. Only the attacks considered in this paper is selective forward and on-off attack. Hence only SF attack is investigated in this paper.

4. Simulation Settings

In order to evaluate the CTCM approach the simulation results are compared with the obtained results of the existing approach. Here, Network Simulator-2 is used to experiment the CTCM approach using simulation. The pseudo code is written in TCL code and the entire functionality of the CTCM approach is verified. By applying REQ-RES methodology, deploying SD nodes, monitoring nodes and sensor nodes in the network, the distance among the nodes are calculated for estimating the location of the sensor nodes.

There are various metrics used to analyze CTCM performance, they are throughput according to the node and node size, end-2-end, PDR and packet overhead. Throughput is the number of bytes successfully received at the destination per time second. Throughput determines the capability of the application running behind the routing protocols, in terms of network scenario and bandwidth under various conditions. The End-2-End delay measures the amount time taken to traverse the routing path. PDR measures the successfully received packets over the total number of packets sent. PDR says the quality of the application in terms of congestion control. In general the network congestion occurs because of routing overhead. Finally, the control overhead is the proportion of traffic that is expended to control as a fraction of the total traffic. It is calculated as the ratio of the number of control packets processed divided by the total number of data and control packets processed, measures the efficiency of

the routing protocol. The control overhead not only gives a metric for the amount of bandwidth available to data packets, but also indicates whether the latency and packet delivery ratio are compromised because of the network congestion and interference generated from control packets. During the request/response the STD nodes include location information about the nodes and forward in addition to the time. By using the time and location the distance is calculated. The timer class estimates the time during the run time, at what time the REQ is generated and at what time it reaches the sensor nodes. Then at what time the RES is generated and at what time the response reaches the REQ node. At this particular time the Estimate Timer will call the scheduler and estimate the distance. Using the distance and the RES node's location is estimated accurately. If the estimated location is not matched with the node submitted location then the node is assumed as malicious node. During this REQ-RES time, the data packet format is verified. If the node follows the data packet format **Figures 2-4** then the nodes are treated as trusted nodes else assumed as malicious node. Because node belongs to different network follows different data format.

Trust value of a node may change during the network activities like; it may be a sender, receiver or an intermediate node in the route. In this paper, through simulation it is demonstrated that why trust management is more important in WSN. It is assumed that 2% of the nodes fall under selective forward attack where 60% of the packets are forwarded in the wrong route. This situation is simulated which is recognized as malicious node. Also out of 20, 40, 60, 80 and 100 nodes the number of SF nodes are calculated and plotted in a graph shown in **Figure 15**.

In order to compare the existing system and the CTCM system, before and after deploying the CTCM algorithm the simulation is executed. The number of SF node generated is programmatically controlled by verifying the node trust value to decide whether a node is a trusted node or malicious node. **Figure 15** illustrates that the CTCM approach validates the nodes properly so that the number of SF node generation is lesser than the existing approach [5].

Even though detecting and eliminating various attacks in the network, it is essential to consider the improvements in quality of service parameters obtained through the simulation. In this paper throughput, time taken, energy delay and packet delivery ratio are the important QoS parameters are evaluated. The CTCM approach is compared with the existing approach in terms of power consumption against number of nodes deployed in the network. The amount of power consumed is measured in joules and the CTCM approach is spent equal amount of energy than the existing approach. This is great because of each node is investigated by executing three different algorithms in each round of network activity. The amount of power consumed by the CTCM approach is lesser than the existing approach [3]. The energy efficiency can be obtained in two ways by reducing the consumption of energy and saving of energy. It is well known that for each activity of the node an amount of energy is consumed from the total energy of the node (**Figure 16**).

In order to evaluate the performance of the network the number of nodes deployed in the network is changed from 10 to 100. The performance is calculated for 10 nodes, 20 nodes and up to 100 nodes deployed in the network in order to compare it.

The remaining energy is calculated by the Equation (20). The remaining energy of each node after one round is calculated and shown in **Figure 17**. From this Fig it is very clear that the energy saving is high in CTCM ap-

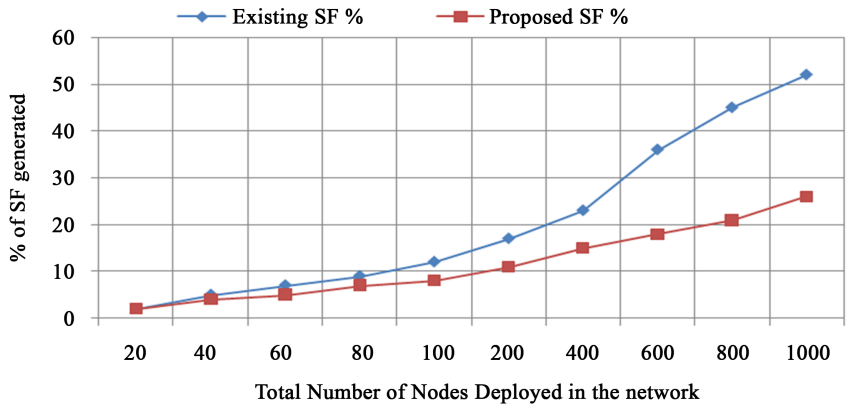

Figure 15. Total number of node deployed vs. % of SF node generated.

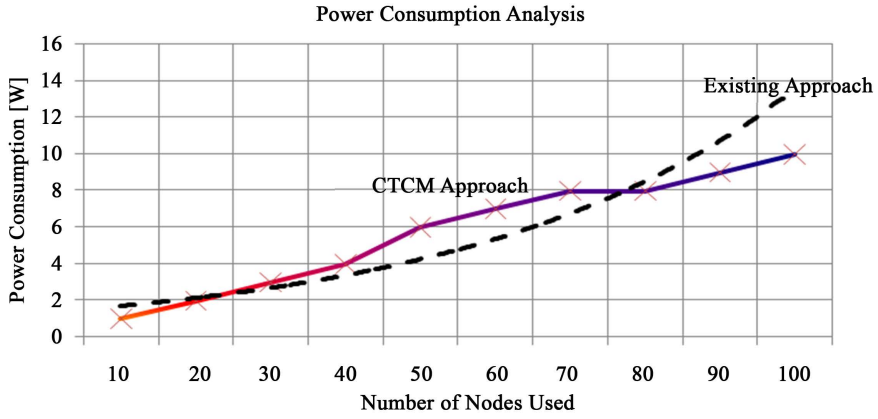

Figure 16. Analysis of energy consumption.

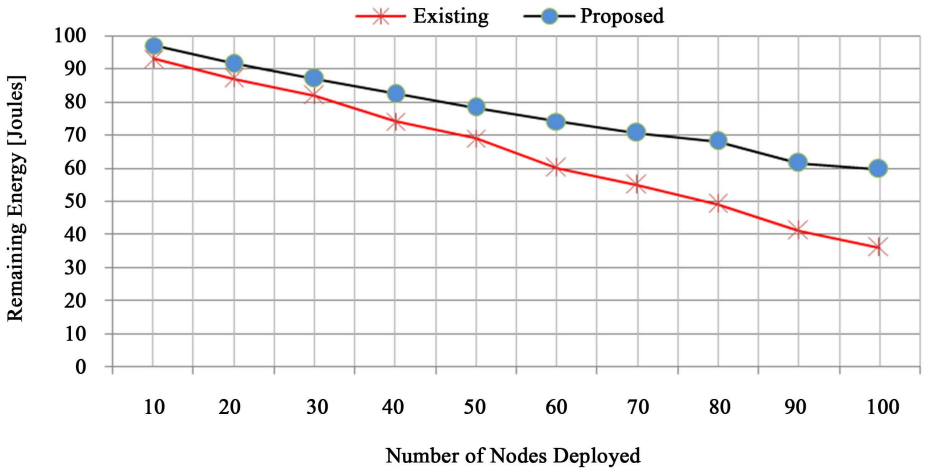

Figure 17. Evaluation in terms of remaining energy.

proach than the existing approach [5]. During the transmission the amount of data successfully transmitted is called as throughput. The constructed path is shortest and the intermediate nodes available in the path are trusted nodes. Because of this throughput obtained by the CTCM approach is better than the existing approach [5]. This improved performance over throughput is achieved by controlling the malicious activity in the network. The obtained throughput comparison between the existing and the CTCM approach is shown in **Figure 18** and it shows that the obtained throughput using CTCM approach is higher than the existing approach [5]. The amount of throughput depends on the number of nodes deployed in the network and the large amount of data transmission. Throughput increases when the number of nodes increased.

One of the main important factors which decide the QoS in WSN is delay. Delay is the time taken to complete one data transmission process, means sending a data packet from one source node to destination node in the network. The amount of delay increases according to the number of nodes in the network, distance among the nodes, and number of intermediate nodes between source and destination node and size of the data transmitted in a route. The amount of delay in terms of number of nodes obtained through the proposed simulation is shown in **Figure 19**. Other QoS parameters are delay and Packet Delivery Ratio (PDR) in the network. Here the delay and the PDR are calculated for one or more rounds of data transmission in the network. The time duration taken for sending data packets from source to destination is termed as delay. Simulation based delay is calculated against number of nodes in one round of network operation is calculated and shown in **Figure 20**. In each round the number of nodes are increased due to that delay is also getting increased. Comparing with the existing, the CTCM approach took less delay for network activities whereas the existing approach took long delay. Within the delay time and the throughput the successful data packets received by the destination node is calculated as PDR. Here the achieved PDR using CTCM approach is higher than the existing approach.

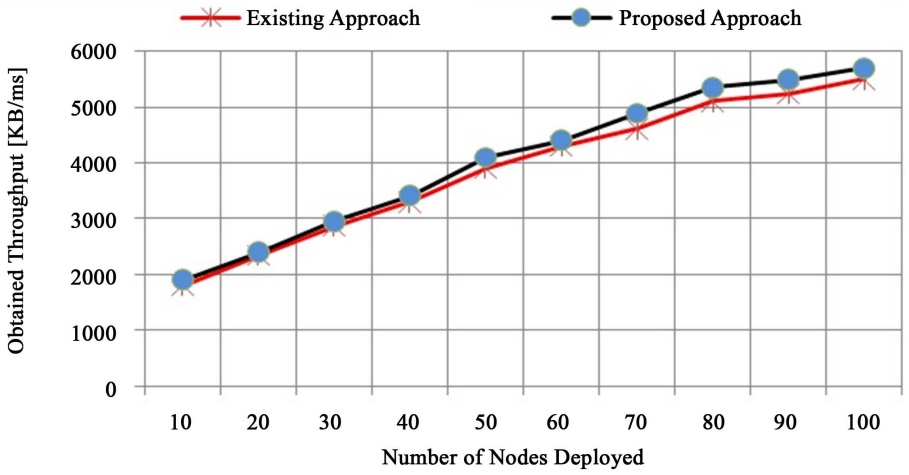

Figure 18. Throughput comparison.

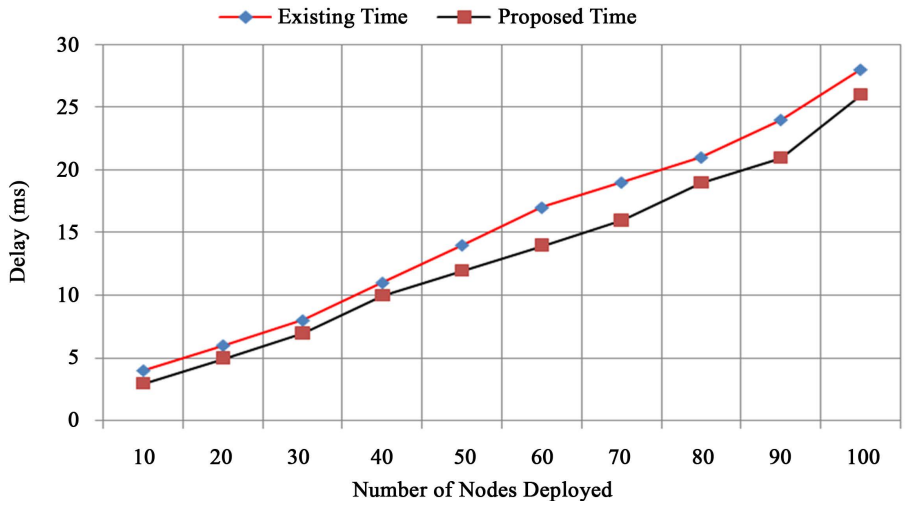

Figure 19. Delay taken for data transmission (for one round).

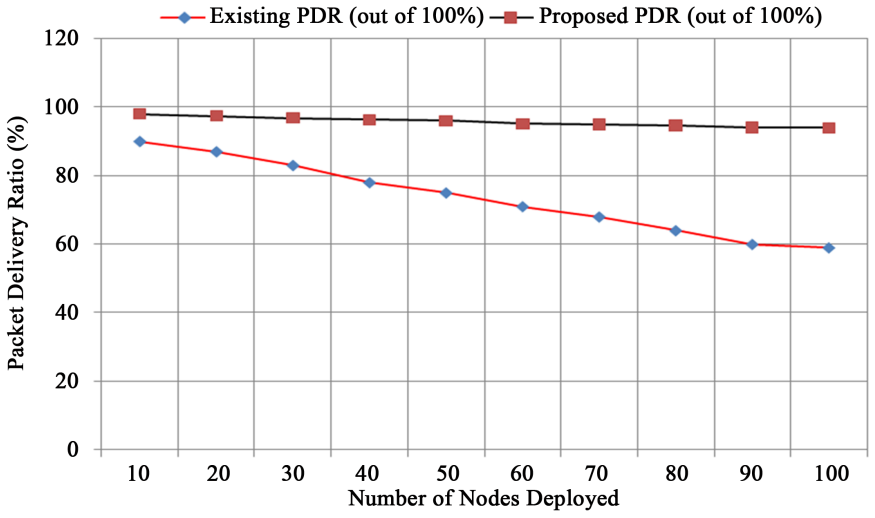

Figure 20. Packet delivery ratio comparison.

5. Conclusion

The main objective of this paper is to provide a trust management based routing mechanism to improve the quality of service in WSN. From the above simulation result it is obtained that the CTCM approach is better than the existing approach in terms of energy consumption, energy saving, throughput, delay and PDR, especially in controlling the malicious activities. Less energy consumption, increased energy saving, increased throughput, less delay and increased PDR give the better performance in the CTCM approach. This improvement is obtained only by choosing trusted nodes, constructing trusted route and trusted data packet transmission. Hence this CTCM approach is a better approach to provide a trust management based routing in WSN. In the future this proposed approach can be applied with clustering protocols to design a better Secured Clustering Routing Protocol for WSN.

References

[1] LeMay, E., Scarfone, K. and Mell, P. (2012) National Institute of Standards and Technology (NIST) Interagency Report 7864, the Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities. http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7864.pdf

[2] Akyildiz, I., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless Sensor Networks: A Survey. *Computer Networks*, **38**, 393-422. http://dx.doi.org/10.1016/S1389-1286(01)00302-4

[3] Lopez, J., Roman, R., Agudo, I. and Fernandez-Gago, C. (2010) Trust Management Systems for Wireless Sensor Networks: Best Practices. *Computer Communications*, **33**, 1086-1093. http://dx.doi.org/10.1016/j.comcom.2010.02.006

[4] Marti, S., Giuli, T., Lai, K. and Baker, M. (2000) Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. *Proceedings of the* 6*th Annual International Conference on Mobile Computing and Networking* (*MOBICOM*'00), Boston, 6-11 August 2000, 255-265. http://dx.doi.org/10.1145/345910.345955

[5] Paul, K. and Westhoff, D. (2002) Context Aware Detection of Selfish Nodes in DSR Based Ad-Hoc Networks. *IEEE Global Telecommunications Conference*, *GLOBECOM*'02, Vol. 1, 178-182.

[6] Buchegger, S. and Le Boudec, J. (2004) A Robust Reputation System for Mobile Ad-Hoc Networks. *Proceedings of P2PEcon*, EPFL IC Technical Report IC/2003/50.

[7] Michiardi, P. and Molva, R. (2002) Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Net-Works. *Proceedings of the IFIP TC*6/*TC*11 6*th Joint Working Conference on Communications and Multimedia Security*: *Advanced Communications and Multimedia Security*, Portorož, 26-27 September 2012, 107-121. http://dx.doi.org/10.1007/978-0-387-35612-9_9

[8] Bansal, S. and Baker, M. (2003) Observation-Based Cooperation Enforcement in Ad Hoc Networks. http://arxiv.org/pdf/cs/0307012.pdf

[9] He, Q., Wu, D. and Khosla, P. (2004) SORI: A Secure and Objective Reputation Based Incentive Scheme for Ad-Hoc Networks. *IEEE WCNC*2004, Vol. 2, 825-830.

[10] Maciá-Fernández, G., Díaz-Verdejo, J., García-Teodoro, P. and deToro-Negro, F. (2008) LoRDAS: A Low-Rate DoS Attack against Application Servers. In: Lopez, J. and Hämmerli, B.M., Eds., *Critical Information Infrastructures Security*, Springer Berlin, Heidelberg, 197-209. http://dx.doi.org/10.1007/978-3-540-89173-4_17

[11] Qin, T., Yu, H., Leung, C., Shen, Z. and Miao, C. (2009) Towards a Trust Aware Cognitive Radio Architecture. *ACM SIGMOBILE Mobile Computing and Communications Review*, **13**, 86-95. http://dx.doi.org/10.1145/1621076.1621085

[12] Cheng, C.-L., Xu, X.-L. and Gao, B.-Z. (2012) ME Trust: A Mutual Evaluation-Based Trust Model for P2P Networks. *International Journal of Automation and Computing*, **9**, 63-71. http://dx.doi.org/10.1007/s11633-012-0617-5

[13] Shila, D.M., Cheng, Y. and Anjali, T. (2010) Mitigating Selective forwarding Attacks with a Channel-Aware Approach in WMNs. *IEEE Transactions on Wireless Communications*, **9**, 1661-1675. http://dx.doi.org/10.1109/TWC.2010.05.090700

[14] Wang, J., Liu, Y. and Jiao, Y. (2011) Building a Trusted Route in a Mobile Ad Hoc Network Considering Communication Reliability and Path Length. *Journal of Network and Computer Applications*, **34**, 1138-1149. http://dx.doi.org/10.1016/j.jnca.2010.11.007

[15] Chang, B.-J. and Kuo, S.-L.(2009) Markov Chain Trust Model for Trust Value Analysis and Key Management in Distributed Multi-Cast MANETs. *IEEE Transactions on Vehicular Technology*, **58**, 1846-1863. http://dx.doi.org/10.1109/TVT.2008.2005415

[16] Liu, Z., Yau, S.S., Peng, D. and Yin, Y. (2008) A Flexible Trust Model for Distributed Service Infrastructures. 11*th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing* (*ISORC*), Orlando, 5-7 May 2008, 108-115. http://dx.doi.org/10.1109/isorc.2008.84