Reputable Path Recognition in Wireless Sensor Network using Intelligent Water Drop

Heena Rathore
Indian Institute of Technology, Jodhpur

Manish Goyal
Infosys, Bangalore

ABSTRACT
Applying trust in Wireless Sensor network (WSN) is an emerging area where researchers are engrossed in developing novel design archetype to address the security issues. Security plays an important role in WSN where trustworthy sensor node turns into a untrustworthy one because of hardware and software faults. Aim of the paper is to propose an optimized path coverage algorithm with the help of an emerging biologically inspired technique, i.e., intelligent water drop (IWD). The proposed work is guaranteed to have most reputable path leading to most trustworthy nodes in WSN. This approach uses dynamic parameters for finding all the optimal paths using basic properties of natural water drops. It proposes how different test cases can be considered as an IWD moving on the edges of the network for finding the optimal paths. The algorithm guarantees complete path coverage.

Keywords
Biologically inspired, Trust, WSN, IWD.

1. INTRODUCTION
Exploring the symbiotic nature of biological systems can result in valuable knowledge for computer networks [2]. Biologically inspired techniques seem promising because of their intrinsic appealing characteristics as self-healing, self-adapting, and self-evolving [3]. They are apter because the techniques are more robust at the time of catastrophic errors when optimized systems fail to deliver optimum performance. While closely looking at nature, we can derive inspiration from everything we see [4]. May it be from animals, birds, waterfall, etc. Birds' flock together synchronizing their pattern, bees perform nectar search with a well-defined procedure, and ants find their food source with the stigmergy. All these social insects along with numerous examples have been studied thoroughly and are used in deriving inspiration in various fields.

Wireless Sensor Network is a network of remote stations /nodes sending sensing information to the gateway. The nodes have the capability to sense various environmental parameters such as temperature, pressure, humidity, etc. However, there are times when the sensor nodes knowingly/ unknowingly send bad information to the gateway. This is because the nodes are remotely located and anybody can hamper the device. For this reason, there is a requirement to look at the security issues of WSN. It is better to find most trustworthy/benevolent nodes in the network, on which the system can rely at its best. The objective of the paper is to present a novel approach for finding the most reputable path leading to trustworthy nodes in the network. Section II gives the related work in the field of trust. Section III is the novel approach that can be used in WSN for recognition of reputable path. Section IV summarizes the paper and presents the scope of future plans.

2. BACKGROUND
In WSN, nodes work selflessly to achieve a common goal like measuring the temperature parameter using thermocouple sensor nodes. Also, the architecture of WSN can be either as flat configuration or clustered configuration as shown in Figure 1.

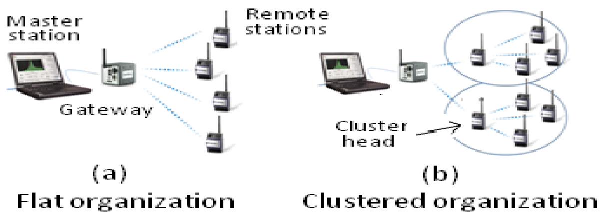

Fig 1: Different architectures of WSN

Nodes can either send the information directly to the gateway or to the cluster heads /relay nodes which in turn sends it to the gateway. Hence the topology can be considered as a tree structure having a source node which senses the information, processes it and sends it the respective destination node.

Sensor nodes precisely have two main functionalities besides processing data, i.e., sensing and communicating data. Both at the time of sensing as well as communicating data, node can misbehave. While sensing the data, the sensor node can misinform by becoming a faulty node because of noise, environmental harsh conditions, or may be because of interference. And while communicating the data, the sensor node can become malicious and become selfish to exhaust nearby node energy as compared to its own. It can be malicious while forwarding or routing packets or it can self exclude data by not sending the data to the base station. This scenario makes it ardent to apply trust mechanisms to WSN to find the trustworthy nodes of the network.

There are various mechanisms used to find the trustworthiness of the network. Momani et al. present a survey of trust models in different network domains. Several techniques such as ratings, weightings, probability, Bayesian network approach, game theory approach, swarm intelligence, neural network and fuzzy logic are used in assessing trust factor of sensor nodes [16]. Swarm intelligence is the bio-inspired technique which uses ant colony optimization, where ants build paths which fulfill certain conditions in a graph. These ants leave some pheromone traces that help next ants to discover and follow those routes [1]. The resultant is that we find the most reputable path. Besides this there are other techniques which were developed, like, ATRM, QDV, ATSN, RFSN, CORE, DBRTS [5], [6], [7], [8], [9], [10].

3. BIO-INSPIRED, INTELLIGENT WATERDROP TECHNIQUE

3.1 Assumptions

There are two status of WSN, either it can be static one, where the nodes are strict to a place or it is a dynamic one where nodes are mobile. There are nodes which remain idle /asleep most of the time or there are nodes which are highly active providing some service. We have assumed a topology where there is high-sensor activity, where there are nodes asking for services and there are nodes providing those services.

We will also assume that every node will only know its neighbors, that is, those nodes within its wireless range. Our model is aimed to help a node requesting a certain service to the network to discover the most trustworthy route leading to a node providing the right requested service. A node (equally a path) can be considered untrustworthy either because it intentionally provides a fraudulent service or because it provides a wrong one due to hardware failures or performance deterioration.

3.2 Intelligent Water Drop

IWD algorithm has been studied extensively which is nothing but a idealized upon studying natural water drops in rivers [13]. Natural water drops have two properties that are soil and velocity. Every water drop flows with some velocity and carries an amount of soil with itself. The velocity of the water drop helps in removing some soil from the riverbeds, and this soil is added to the soil of the water drop. During the transition, the water drops gains speed; due to which a water drop with more velocity can erode more soil. One more interesting feature of the water drop is that it chooses an easy path when there are many paths available at the same time. Here, easy path means the path with lesser soil. In this way constructively, the water drop chooses its optimal path. Using these remarkable properties IWDs are created. These drops possess two important properties of natural water drops that are:

1. the soil it carries, denoted by soil (IWD)
2. the velocity that it possesses, denoted by velocity (IWD).

Both these values can change when an IWD flows from one location to another in its environment (given problem statement). The IWDs' velocity is non-linearly proportional to the inverse of the soil on the path between the two locations. The IWDs' soil is increased by removing some soil from the path, and the amount of soil added is non- linearly proportional to the inverse of the time taken to move between the locations. This time is proportional to the velocity and inversely proportional to the distance between the two locations. In addition, the path choosing property is implemented by calculating an uniform random distribution function, such that the probability of choosing a path is inversely proportional to the soil of the available paths. This paper briefs about IWDs, which is used to find optimized path of WSN to find the most trustworthy nodes in the network. The following section explains how the properties of water drops are applied to the desired problem and the process of achieving the reputable paths.

3.3 Proposed Work

We presume to know the topology of the network at the start. Initially, the important factors affecting the working of the algorithm are explained followed by a stepwise explanation of the algorithm.

1. Fitness function

A fitness function has been designed for the calculation of soil on all the links of the graph [13]. The basic idea behind using the fitness function is to give importance to more deep and complex paths. A well-constructed fitness function may increase the chances of finding a solution and reaching higher coverage, which is always desirable. For the proposed algorithm, the fitness function can be calculated as:

soil (i, j) =a*subgraph (j) +b*condition (j)

where soil (i,j) is the soil on the edge between the two nodes i and j. Condition (j) of a node j can be either 0 or 1, depending on whether a particular node has a decision element or not. If it has a decision statement then it is assigned a value of 1 else 0. Subgraph (j) is the subgraph value, i.e., number of nodes below that particular node (j) in a given graph. The IWD starts from the initial node of the graph and after that the probability of each path is calculated using the above mentioned fitness function. The path with the highest probability is considered as the desired path for the water drop. The configuration parameter 'a' and 'b' is assigned a constant value of 2 and 1 respectively. The reason behind assigning these values is that, more weight should be assigned to nodes with a larger subgraph value rather than nodes with smaller subgraph value. Subgraph value has been given greater importance in the designing of the fitness function, because by traversing the part of the code with greater depth the probability of achieving complete code coverage is high. This fitness function was so chosen, because during code coverage there are chances of errors in either decision nodes or more dense and complex paths. In order to assign weight to links based on their complexity, this fitness function is chosen.

2. List of visited paths

A list of visited paths known as 'Vc' is maintained to record the paths traversed by the IWD. In order to keep track of the nodes visited, a check is performed on the node, and if all its children are visited then it is deleted. This operation is performed recursively. This is done in order to avoid the IWD from traversing the path which has been already traversed.

3. Constant values

Various configuration parameters (a_s, b_s, c_s), (a_v, b_v, c_v), ρ are used in the calculation and updating of soil and velocity respectively during implementation of the IWD algorithm [11], [12].

3.4 Algorithm

Parameters which are used are:

- Probability (i, j): probability of the IWD to move from node i to node j.
- Time (i, j): time taken by the IWD to move from node i to node j.
- ΔSoil (i, j, iwd): soil eroded by the IWD on the link between node i and node j.
- Vel (i, j, iwd): velocity of the IWD when it moves from node i to node j.

Step wise implementation of the algorithm is [15]:

Algorithm 1: Bio-inspired Intelligent Water drop algorithm

STEP 1: INITIALIZATION OF PARAMETERS

a. Velocity updating parameters are a_v, b_v, c_v. set as: $a_v = 1$, $b_v = 1$, $c_v = 1$.

b. Soil updating parameters are a_s, b_s, c_s set as: $a_s = 1$, $b_s = 1$, $c_s = 1$.

c. The local soil updating parameter is ρ. Here ρ = 0.1 is chosen for better experimental results. Value of ρ can range from 0 to 1, depending upon the amount of soil on the link (nodes).

d. The initial soil on each edge of the graph is computed using the fitness function and the configuration parameters

e. The initial velocity of each IWD is set to 100.

f. Every IWD has a visited path list Vc (IWD) which is initially empty i.e. Vc (IWD) = {}.

STEP 2: UPDATION OF PARAMETERS VALUES

Repeat below steps until topology does not gets empty:

2.1. Calculation of probability:

For the IWD receding at node i, the next node j (which is not in visited node list Vc (IWD) is chosen on the basis of probability $p_i^{IWD}(j)$. The probability for IWD to move from the current node i to node j is calculated using:

Probability (i, j) = Soil (i, j) / Σ k ≠vc Soil (i, k)

Where: Soil (i,j) is the soil between the two nodes, Σ k ≠vc Soil (i,k) denotes the summation of the soils of all the paths which can be traversed from the current node i. In the original algorithm, the IWD moves to the location where the soil is less, but in our case we traverse the IWD to the path with more soil.

2.3 Calculation of time taken:

After calculating the probability, IWD chooses the next node which it has to move on. After selecting the respective node, the time taken to move from node i to node j is calculated as:

Time (i.j) = (subgraph (i)–subgraph (j)) / vel (iwd)

Where subgraph(i) – sugraph(j) correspond to the distance between the nodes and vel(iwd) is the original velocity of the IWD.

2.4 Calculation of Soil of the IWD:

Soil carried by IWD is computed using 2.3:

ΔSoil (i, j, iwd) = a_s / (b_s+c_s*time (i, j))

Where: a_s, b_s, c_s are the positive parameters as specified earlier.

2.5 Calculation of Velocity of the IWD:

The velocity of the IWD after it has moved from node i to j is calculated as:

vel(i,j,iwd) = vel(iwd) +(a_v +(b_v+c_v soil(i,j)))

Where: soil (i,j) is the soil on the path before the IWD traversed the required path. a_v ,b_v, c_v are the positive parameters as specified earlier and vel(IWD) is the previous velocity of the IWD.

2.6 Soil updation of the link:

Since, the IWD carries some amount of soil with it; the soil on the link is reduced. Therefore, the soil on the link is updated as:

Soil (i,j)= (1-ρ) soil (i,j) – ρ * Δsoil (i,j,iwd),

Where: ρ is a positive parameter, Δsoil (i, j, iwd) is the soil carried by the IWD while moving from the node i and j.

2.7

Current node is updated and the same procedure from step 2.2 is repeated until it does not reach leaf node, or one complete path is not reached. After it reaches the leaf node then that node is deleted, and its parent is also checked. If all its leaves have been deleted, then it is also deleted.

2.8

After completing one possible path by the water drop, the visited path list is updated and the IWD again starts with a new iteration where all parameters are initialized again from step 1 onwards. If the topology becomes empty, that means all the paths have been traversed, and we have to exit the loop else all the steps from 2.1 are executed.

In this way, all the possible paths are determined with their corresponding weights. The path with highest weight should be the most trustworthy path. Next section shows the demonstration of the above mentioned algorithm on an illustrated problem.

The complete flowchart of the algorithm is:

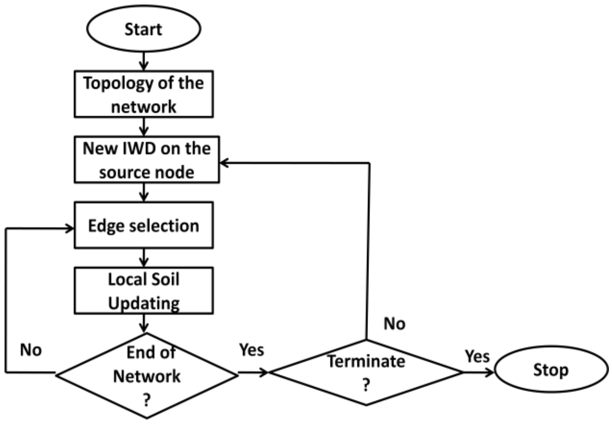

Fig 2: Flowchart for the IWD algorithm

Next we describe the above algorithm with the help of example:

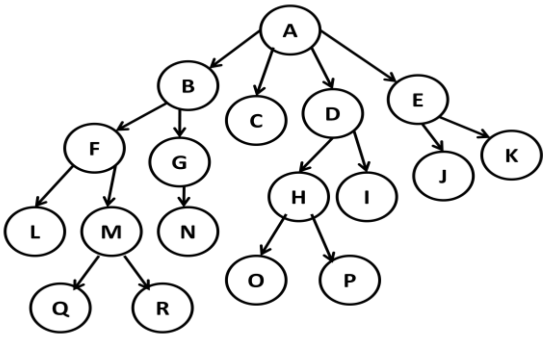

Fig 3: Topology of the sensor network

In this topology, A can be considered as the source node from where data is transmitted. Sub-graph of all the nodes is calculated first. Sub-graph value of a node is the number of nodes below that particular node. Node A has all the nodes, i.e., B, C, D, and so on connected to it. This is done using the depth first traversal algorithm [14]. Hence the value of the sub-graph of all nodes is:

A = 17, B=7, D=4, E= 2, F=4, G=1, H=2, M=2

C, I, J, K, L, N, O, P, Q and R have a sub graph value of 0.

Now, the decision condition for the node is checked. A, B, D, E, F, G, H and M are decision nodes so they get 1 and rest of them get a value of 0. The various iterations of the IWD are shown below.

Iteration 1: The IWD starts from starting node A as shown in Figure 4.

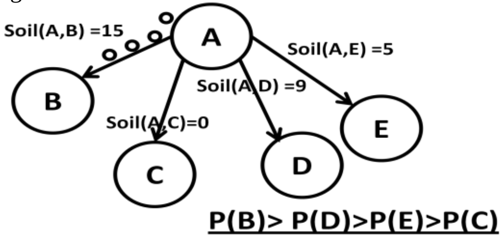

Fig4: Path selection by IWD from A

Note: Visited path (Vc)= A->B

As the water-drop, have four paths to travel, i.e., via B, C, D and E. So, in order to find out which node to be traversed next, the probability of two nodes is computed using the respective link soil and on the basis of the next node for the 1st path is selected.

The calculation of the various parameters is

Soil (A, B) =2*7+1*1=15

Soil (A, C) =2*0+1*0=1 (as sub-graph(j)&&condition(j)==0)

Soil (A, D) =2*4+1*1=9
Soil (A, E) =2*2+1*1=5
P (A, B) = Soil (A, B)/ {Soil (A, B) + Soil (A, B) +Soil (A, B) + Soil (A, B)}
= 15/30
P (A, C) = 1/30
P (A, D) = 9/30
P (A, E) = 5/30
The node with a higher probability is chosen first. In this case probability of node B is higher than that of node C, D and E so node B is selected. The probability is calculated using the Step 2.2 of the algorithm mentioned above. The time taken by the IWD to move from A–B is calculated using Step 2.3 of the algorithm mentioned above:
Time (A, B) = max {(17-7), 1}/Velocity (Viwd)
=10/100 =0.1 sec
The soil carried by the IWD and its velocity is computed which is then subtracted from the soil of the link A-B to compute the remaining soil on the link using steps 2.3 and 2.4 and 2.5:
ΔSoil (A, B) = 1/ (1 +1*.1) = 0.9
ΔVelocity (A, B) = 100 + 1/1+ 1*15= 100.0625
Updated Soil (A, B) = 0.9*15–0.1*0.9= 13.41
Soil (iwd) = 0.9
So the path will be: A – B.
Now, after reaching node B the water-drop has two available nodes for selecting the next node, which is shown in Figure 5.

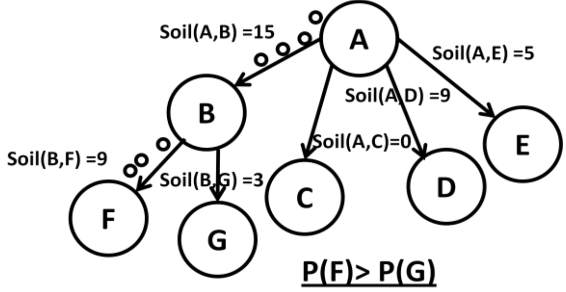

Fig5: Path selection by IWD from B

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Again, the probability of all the available nodes is calculated and on the basis of that the next node is chosen.

Soil (B, F) =2*4+1*1=9

Soil (B, G) =2*1+1*1=3

P (B, F) = Soil (B, F)/ {Soil (B, F) + Soil (B, G)}

= 9/12=0.75

P (B, G) = 0.25

As the probability of node F is greater so the water drop will flow to node F.

Time (B, F) = {7-4}/100.0625= 0.0299

ΔSoil (B, F) = 1/ (1 +1*.0299) = 0.9709

ΔVelocity (B, F) = 100.0625 + 1/1+ 1*9= 100.1625

Updated Soil (B, F) = 0.9*0–0.1*0.970= 8.003

Soil (iwd) = 0.9+0.970=1.87

So the path will be: A–B – F.

Similarly, after reaching node F the water-drop has to decide the next node to be traversed as in Figure 6.

The calculations for the various parameters are shown below:

Soil (F, L) =2*0+1*0=1

Soil (F, M) =2*2+1*1=5

P (F, L) = Soil (F, L)/ {Soil (F, L) + Soil (F, M)}

=1/6=0.167

P (F, M) = 5/6=0.83

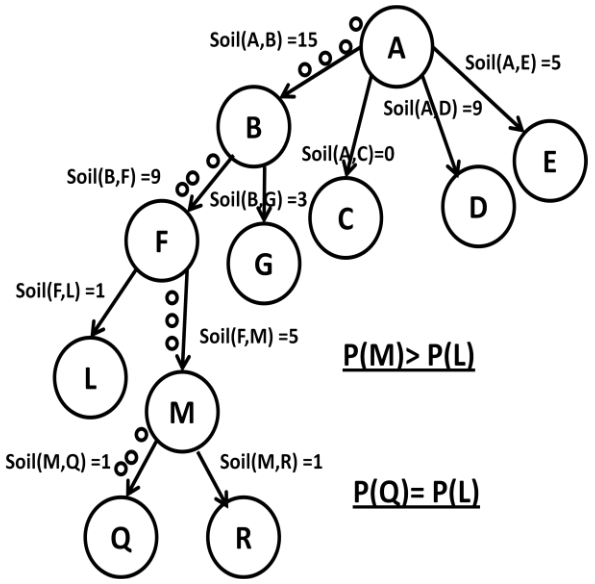

Fig 6: Path selection by IWD from F and M

As the probability of node M is greater so the water drop will flow to node M

Time (F, M) = {4-2}/100.1625= 0.01996

ΔSoil (F, M) = 1/ (1 +1*.01996) = 0.98

ΔVelocity (F, M) = 100.1625 + 1/1+ 1*5= 100.329

Updated Soil (F, M) = 0.9*5–0.1*0.98= 4.402

Soil (iwd) =1.87+ 0.98=2.85

So the path will be: A–B –F–M.

Soil (M, Q) =2*0+1*0=1

Soil (M, R) =2*0+1*0=1

P (M, Q) = P (M, R) = 1/2=0.5

Now, as the probability of two available nodes is same. A random selection is to be done among these three nodes. So, node Q is randomly chosen as the next node.

Time (M, Q) = {2-0}/100.329= 0.01993

ΔSoil (M, Q) = 1/ (1 +1*.0199) = 0.98

ΔVelocity (M, Q) = 100.329 + 1/1+ 1*1= 100.829

Updated Soil (M, Q) = 0.9*1–0.1*0.98= 0.802

Soil (iwd) =2.85+0.98=3.83

So the path will be: A–B –F–M–Q

Now, the calculations for the second iteration for the IWD are shown below. The water drop starts gain from the root of the node i.e. node A.

Iteration 2:

P (A, B) =13.41/28.41=0.472

P (A, C) = 1/28.41=0.351

P (A, D) = 9/28.41=0.316

P (A, E) = 5/28.41=0.1759

Since probability of node B is more we traverse to node B.

Time (A, B) = max {(17-7), 1}/Velocity (V_{iwd})

=10/100 =0.1 sec

ΔSoil (A, B) = 1/ (1 +1*.1) = 0.9

ΔVelocity (A, B) = 100 + 1/1+ 1*13.41= 100.069

Updated Soil (A, B) = 0.9*13.41–0.1*0.9= 11.979

Soil (iwd) = 0.9+3.83=4.73

So the path will be: A – B.

P (B, F) = Soil (B, F)/ {Soil (B, F) + Soil (B, G)}

= 8.003/11.003=0.727

P (B, G) = 0.272

As the probability of node F is greater so the water drop will flow to node F.

Time (B, F) = {7-4}/100.069= 0.029

ΔSoil (B, F) = 1/ (1 +1*.029) = 0.971

ΔVelocity (B, F) = 100.069 + 1/1+ 1*8.003= 100.180

Updated Soil (A, B) = 0.9*8.003–0.1*0.971= 7.1056

Soil (iwd) = 4.73+0.970=5.701

So the path will be: A–B – F.

Similarly, after reaching node F the water-drop has to decide the next node.

The calculations for the various parameters are shown below:

P (F, M) =1/5.402=0.814

P (F, L) = 1/5.402=0.185

As the probability of node M is greater so the water drop will flow to node M

Time (F, M) = {4-2}/100.180= 0.0199 sec

ΔSoil (F, M) = 1/ (1 +1*.0199) = 0.9804

ΔVelocity (F, M) = 100.180 + 1/1+ 1*4.402= 100.365

Updated Soil (F, M) = 0.9*4,402–0.1*0.98= 3.863

Soil (iwd) =5.701+ 0.98=6.814

So the path will be: A–B –F–M.

Now,

Time (M, R) = {2-0}/100.365= 0.0199 sec

ΔSoil (M, R) = 1/ (1 +1*.0199) = 0.98

ΔVelocity (M, R) = 100.365 + 1/1+ 1*1= 100.865

Updated Soil (M, R) = 0.9*1–0.1*0.98= 0.802

So the path will be: A–B –F–M–R

In a similar manner, we apply the algorithm and compute various paths. The path with the maximum weight indicates the most trustworthy path.

Path 1: A → B → F → M →Q (Total weight of path is: 30)

Path 2: A → B → F → M →R

(Total weight of path is: 26.617)

Path 3: A → B → F → L (Total weight of path is: 22.9476)

Path 4: A → B →G →N (Total weight of path is: 17.9881)

Path 5: A → D →H →O (Total weight of path is: 15)

Path 6: A → D →H →P (Total weight of path is: 9.6476)

Path 7: A → D →I (Total weight of path is: 7.99)

Path 8: A → E →J (Total weight of path is: 6)

Path 9: A → E→K (Total weight of path is: 5.21)

Path 10: A → C (Total weight of path is: 3.88)

This data can be shown in Table 1.

Table 1: Path strength table

Path No	Path	Weight	Priority
1	A → B → F → M →Q	30	1
2	A → B → F → M →R	26.617	2
3	A → B → F → L	22.9476	3
4	A → B →G →N	17.9881	4
5	A → D →H →O	15	5
6	A → D →H →P	9.6476	6
7	A → D →I	7.99	7
8	A → E →J	6	8
9	A → E→K	5.21	9
10	A → C	3.88	10

The proposed algorithm uses IWD concept to generate all the possible independent paths with their priority, which is depicted in the path strength in Table 1. The weight of each path is calculated by adding the soil on all the links of the path. It shows that the path with highest weight has the highest priority. The beauty of the algorithm is that it traverses all the paths as well as the nodes and computes the longest and the best path. Weight can be said as the trust value of that path and ΔSoil (i, j) is the trust of a particular communication between node i and j.

4. Conclusions

Managing trust in Wireless Sensor Networks (WSNs) is a crucial issue and necessary steps should be taken for security purpose in an efficient, accurate and robust way. Providing this management would notably increase the security in such a sentient environment, supporting thus its development and deployment.

In this paper, we have proposed a new method which is biologically inspired and is used to find the most reputable path leading to find the trustworthy nodes in the network. The topology here used is a tree topology, however we can apply this algorithm to any of the network graph structure where there are nodes asking for the services known as source nodes and there are nodes providing certain services as destination. Specifically we have explained how to measure the quality of the path. This in turns computes the trust value of a particular link.

5. REFERENCES

[1] Gomez Marmol, Felix, and Gregorio Martínez Perez. ``Providing trust in wireless sensor networks using a bio-inspired technique." *Telecommunication systems*, vol. 46, no. 2, pp. 163-180, 2011.

[2] Heena Rathore, Sushmita Jha, ``Bio-Inspired Machine Learning Based Wireless Sensor Network Security", *IEEE World Congress on Nature and Biologically Inspired Computing*, no. 5, pp. 140 - 146, 2013.

[3] Heena Rathore, Venkataramana Badarla, Sushmita Jha, Anupam Gupta. ``Novel Approach for Security in Wireless Sensor Network using Bio-Inspirations", in Proc. *IEEE International Conference on Communication Systems and Networks* (COMSNETS), no. 6, pp. 1-8, 2014.

[4] Heena Rathore, Abhay Samant, ``A system for building immunity in social networks", in proc. *World Congress on Nature and Biologically Inspired Computing (NaBIC),* no.4, pp. 20-24, 2012.

[5] Boukerche, A., Xu, L., & El-Khatib, K.. ``Trust-based security for wireless ad hoc and sensor networks''. *Computer Commu- nications*, vol. 30, no. 11–12, pp. 2413–2427, 2007.

[6] Dhurandher, S. K., Misra, S., Obaidat, M. S., & Gupta, N. ``An ant colony optimization approach for reputation and quality- of-service-based security in wireless sensor networks''. *Security and Communication Networks*, vol. 2, no. 2, pp. 215–224, 2009.

[7] Chen, H., Wu, H., Zhou, X., & Gao, C. ``Agent-based trust model in wireless sensor networks'', in proc. *ACIS international conference on software engineering, artificial intelligence, net- working, and parallel/distributed computing*, SNPD'03 no.8, pp. 119– 124, 2007.

[8] Ganeriwal, S., & Srivastava, M. B. ``Reputation-based framework for high integrity sensor networks'', in proc. *ACM workshop on security of Ad hoc and sensor networks*, no. 2, pp. 66–77, 2004.

[9] Michiardi, P., & Molva, R. (2002). ``CORE: a collaborative reputa- tion mechanism to enforce node cooperation in mobile Ad hoc net- works'', in proc. *IFIP TC6/TC11 sixth joint working conference on communications and multimedia security*, pp. 107– 121, 2002.

[10] Srinivasan, A.,Teitelbaum, J., & Wu, J. ``DRBTS: distrib- uted reputation-based beacon trust system'', in proc. *IEEE International Symposium on dependable, autonomic and secure computing*, no. 2, pp. 277–283, 2006.

[11] Shah-Hosseini, H. ``Problem solving by intelligent water drops'', in proc. *IEEE Congress on Evolutionary Computation*, IEEE Computer Society, pp.3226–3231, 2007.

[12] Shah-Hosseni, H., ``Optimization with the nature-inspired intelligent water drops algorithm'', *International Journal of Bio-Inspired Computation*, vol. 1, nos. 1/2, pp.105–125, 2009.

[13] Chen, Y. et al., ``Comparison of two fitness functions for GA-based path-oriented test data generation'', in proc. *International Conference on Natural Computation*, IEEE Computer Society, no.5, pp.2–4, 2009.

[14] Cormen, T.H. et al. Introduction to Algorithms, 2nd ed., MIT Press, USA. 2001.

[15] Komal Agarwal, Manish Goyal, Praveen Ranjan Srivastava, `` Code Coverage using Intelligent water drop'', *International Journal of Bio-inspired Computation*, vol. 4, no. 6, pp 392-402, 2012.

[16] Mohammad Momani , Subhash Challa,`` Survey of Trust Models in Different Network Domains'', *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, vol. 1, no.3, pp. 1-19, 2010.