

# Energy Efficient Cluster Head Selection In Software Defined Networking Using Improved Particle Swarm Optimization

S.Suganthi, Dr.D.Usha

**Abstract:** In real world applications internet of things becoming an important role. Most IoT applications are integrated with wireless sensor networks. Usually the wireless sensor networks in IoT require hundreds or thousands of sensors may be deployed and integrated. In this scenario management of networks is the biggest issue. To manage the larger network scenario software defined networking is an added advantage. It gives a promising solution for flexible management of data plane and control plane. Efficient transmission of data with minimum energy is the main goal. So dividing the nodes into multiple clusters and cluster head is needed for manage those clusters. To maximizing the lifetime of the network with minimum energy there is in need of energy efficient cluster head selection. We provide the optimality in cluster head selection by using particle swarm optimization. There are so many researchers are already done this work with PSO but the results are not up to the level. This paper demonstrated the updated PSO algorithm through modified and improved fitness function. The proposed algorithm is experimented in matlab and the results are evaluated to show their supremacy in term of alive nodes, energy expenditure, dead nodes and fitness value.

**Index Terms:** Sensor network, clustering algorithms, energy, cluster head, particle swarm optimization.

## I. INTRODUCTION

In recent years, the wireless sensor networks become a leading area of research. The device in sensor networks are denoted as sensor which has detect and respond some input from both physical as well as environmental conditions such as heat, pressure etc. the output of the sensor usually an electrical signal which is transmitted to a controller for further processing. For managing and maintaining larger networks the software defined networking is the biggest boon. In a software defined networking the networks are divided into control plane and data plane. Control plane is the centralized control which shapes the network traffic without touch the individual switches. Data plane can carry on the forwarding activity based on the flow table which is pre-designed by the controller. So through SDN can get better security because of the controller can monitor traffic and deploy security policies and also the controller detects suspicious activity in network traffic, it can reroute or drop the packets. Sometimes the sensor nodes in unattended hostile region recharge of sensor battery is not possible. There are so many energy preserving approaches are available which is mainly focused to extend the network lifetime. To support scalability and lifetime of the network, nodes are often grouped in clusters having a leader, often referred as cluster head (CH). A CH is responsible for not only sending data to base station but also assist the general nodes to send sensed data to target nodes.

The energy consumption of CH is greater than general nodes. Cluster head is a node which is dependable for maintain cluster, collect data from nodes in the cluster and communicate with sink. By using clustering method it has been observed that there is huge quantity of energy that has been saved. Therefore CH selection will affect the lifetime of WSN. The paper proposes an approach for improving network lifetime by using Particle swarm optimization (PSO) in WSN. Particle Swarm Optimization (PSO) is an artificial intelligence technique which is motivated by the social activities of natural species, for instance - swarm of birds, etc. The rest of this paper is organized as follows: the network model and assumptions are briefly outlined in Section II. In Sections III proposed PSO algorithm the simulation results are discussed in section IV. Finally, a projection of future PSO applications in WSNs and concluding remarks are given in Section V.

## II. CLUSTER FORMATION

In this section initially we describe our network scenario. The SDN network is assumed to be a circular geographic region with the coordinates (X,Y) in which the control plane and data plane is separated. The sensor nodes are uniformly deployed in the data plane area  $dA$ . The controller console manages the policies of each node in a data plane which is denoted as  $cA$ . The initial energy of each node is  $E_0 = 200$ . Weight of a node  $w = 0.9$  Cluster radius denoted as  $C_r = 30$  and sensing range is  $S_{ra} = 36$ . Each node broadcasting the beacon message to all other neighboring nodes about their presence. The beacon message contains the node id, current region id, cluster membership status, radio range and velocity. If a node  $N_j$  receives a beacon message from node  $N_i$ , the region id and radio range is same then  $N_j$  add  $N_i$  to form as a cluster. Then the cluster information is automatically updated in node information table which is maintained by controller. The node information table comprises the ids of all nodes along with their reputation values. Initially

- 
- S.Suganthi, Dr.D.Usha Research Scholar, Assistant Professor2,
  - Mother Teresa Women's University, Kodaikanal

thereputation value of all the nodes is to some default value. The reputation value is based on the energy and distance among the clustering nodes which is calculated by using non linear programming. If the reputation value  $R_v$  falls below the threshold then the corresponding node will automatically search other node in another cluster in its neighborhood.

**Algorithm 1: Cluster Formation Algorithm**

<p>Coordinates: (X, Y), N - Set of all sensor nodes, <math>C_p</math> : Control Plane,</p> <p><math>D_p</math> : Data Plan e,</p> <p><math>Cid</math> : Cluster id,</p> <p><math>B_m</math> : Beacon Message,</p> <p>NIF : Node Information Table, <math>E_0</math> : Energy of node,</p> <p><math>W</math> : Weight</p> <p><math>C_r</math> : Cluster radius,</p> <p><math>R_{ra}</math> : Radio range,</p> <p><math>S_{ra}</math> : Sensing range, <math>V</math> : Velocity,</p> <p><math>R_{id}</math> : Region id,</p> <p><math>R_v</math> : Reputation value,</p> <p><math>d_{ij}</math> : the distance from sensor node <math>i \in N</math> to sensor node,</p> <p><math>f_i</math> : the distance from sensor node <math>i \in N</math> to the Base station BS,</p> <p><math>b_j</math> : the battery level of sensor node <math>i &lt; n</math></p>	<ol style="list-style-type: none"> <li>1. <math>N \leftarrow 100</math> ;</li> <li>2. <math>B_m \leftarrow \{N_{id}, R_{id}, Cluster\ Staus, R_{ra}, V\}</math></li> <li>3. If <math>(R_{id}(N_i) = R_{id}(N_j) \ \&amp; \ R_{ra}(N_i) = R_{ra}(N_j))</math> then</li> <li>4. <math>C_{id} \leftarrow \{ (N_i, N_j) \}</math></li> <li>5. Update <math>C_{id}</math> information to <math>C_p</math></li> <li>6. <math>C_{id} (NIF) \leftarrow \{ N_{id}, R_v \}</math></li> <li>7. <math>R_v \leftarrow (Avg\{Energy\} \ \&amp; \ Avg\{distance\})</math> Reputation value calculated by non linear programming</li> <li>8. <math>R_{ra} = \{1 \text{ if a node } i \text{ and node } j \text{ is underlying specified radio range } \forall i, j; 1 \leq i \leq N, 1 \leq j \leq M, 0 \text{ otherwise } \}</math></li> </ol> $9. \text{ Average Energy}(C_{id}) \leftarrow \frac{\sum_{i=1}^{N_M} \sum_{j=1}^{M} (E_i R_{ra})}{\sum_{i=1, j=1}^{N, M} E_{ij}(C_{id})}$ $10. \text{ Distance } d_{ij} \leftarrow \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ $11. \text{ Average Distance}(C_{id}) \leftarrow \frac{\sum_{i=1}^N \sum_{j=1}^M (d_{ij} R_{ra})}{\sum_{i=1, j=1}^{N, M} D_{ij}(C_{id})}$ <ol style="list-style-type: none"> <li>12. if <math>(Avg(d_{ij})(C_{id}) &lt; T</math> or <math>(Avg(EC_{id})) &lt; T</math></li> <li>13. search again all the nodes in corresponding cluster id to find another neighborhood node to form a cluster</li> </ol>
---	--

**III. CLUSTER HEAD SELECTION USING PARTICLE SWARM OPTIMIZER**

representing the  $i$ th particle  $P_i$  of the population is as follows:

PSO algorithm is a global optimization algorithm (PSO) based on natural population search strategy, it has advantages of being easy to carry out, less parameters, fast search speed, etc. and is widely used in scientific research and practical engineering. To optimize the performance of particle swarm algorithm, this paper respectively improve 4 aspects, speed, inertia weight, learning strategy and variability, to improve search precision and stability of the algorithm. Particle Swarm can be specifically described as follows: For prolong a lifetime of a network the cluster head selection is very important because of while receiving and aggregating the information the leader node is crucial one. Thus the cluster head election algorithm has an important role for energy conservation of a sensor node. In order to improve the optimality for electing the cluster head we use

the Particle Swarm Optimization (PSO) algorithm. In

conventional PSO algorithm, particles in the swarm update their position relative to the position and velocity of the group. The notation for

$$P_i = [ X_{i1}, X_{i2}, X_{i3}, \dots, X_{iD}]$$

Each particle's position is evaluated using a fitness function which judges the quality of the solution provided in that iteration. The velocity and position of each particle in each iteration  $V_{id}$  and  $X_{id}$  are updated as follows:

$$V_{id}(t) = w * V_{id}(t-1) + c_1 * r_1 * (X_{pbestid} - X_{id}(t-1)) + c_2 * r_2 * (X_{gbestid} - X_{id}(t-1))$$

$W$  = inertial weight,  $c_1$  and  $c_2$  = acceleration factor (non negative constants),  $r_1$  and  $r_2$  = distributed random numbers in the range of  $[0, 1]$ . The process are repeated until  $G_{best}$  is achieved or a fixed number of iterations  $t_{max}$  is reached.

**IV. PROPOSED PSO ALGORITHM**

The input hypotheses are plotted in X and Y coordinates and seeded with an energy, weight, cluster radius, sensing range and velocity. Particles move through the solution space and evaluated some fitness function. In every iteration each particle updated two best values such that pbest and gbest. Each time calculate the current position and velocity. The current position of node assigned as Pbest. The initial utility function as assigned as  $Utility = [v1 \ v2]$ .

**TABLE-I : Initial PSO Parameters**

PARAMETER	V A L U E
No.of Particles	100
Inertia weight	0.9 to 0.4 (linear decreasing)
Max.velocity	50
Learning factors	C1=c2=2
Maximum iteration	300
RE0/ED0* Cn	

Definition 1(Weight) : Let W be the weight of a sensor node, Wmax and Wmin are the maximum and minimum weights of a sensor node. Weight W can

$$Min$$

$$id$$

$$(k) = \lim_{k \rightarrow 1 \text{ to } NP} \begin{cases} \frac{W_{max} - W_{min}}{Sr} & \text{if } Di(k) < Sr \\ 0 & \text{else} \end{cases}$$

act as a role of balancing local and global search it can be positive constant or positive linear. an inertia weight are

linearly reduced to search proceeds from 0.9 to 0.4. Definition 2(Sensing range): Let SR be the sensing range of a sensor node which is computed based on the distance between sensor deployment point and the particle. Which is assigned as  $SR = \forall R \forall i ( \text{Max}(\text{distance}(SR, P_i))$ , SR denoted as sensor deployment point, R is the region, Pj denoted as the

Definition 5 (Energy) : The initial energy of node as 200 Jules. The average energy of sensor nodes can calculated as

$$(RE0/ED0 * ) / Cn$$

E0

Definition 6 (Length) The length is calculated by finding the number of arguments of minimum id that is mentioned as length of the live sensor nodes.

$$\text{Arg}(\text{minid})$$

Calculation of fitness function

particle.

$$Fac1 = \sum_{Cn}^1 Dist$$

Definition 3(Distance): The estimated distance between the sensor nodes Xi1,Xi2,Xi3.....XiD and

// factor 1 depending upon distance between the node

$$(E0 * ) / Cn$$

neighbouring nodes Yi1,Yi2,Yi3.....YiD. The method

Fac2 =

E0

for calculating distance between nodes are follows : dij

$$\leftarrow \sqrt{(xj - xi)^2 + (yj - yi)^2}$$

Definition 4(Index Count): In WSN, the sensor node distances below from the sensing range that node added into index count. If Cn is zero discard that particular index.

**Algorithm 2: PSO Algorithm**

//factor 2 depending upon the energy

Fac3 = arg (Mi )

//factor 3 depending upon the arguments count of

Minid

// finding best position by using above 3 factor

$$Bprebestpos = \lim_{i \rightarrow 1 \text{ to } NP} \alpha_1 Fac1_i + \alpha_2 Fac2_i + \alpha_3 Fac3_i$$

1. Input:  $Nor_{node}$  // Node location (x, y) coordinates
2. Output: CH, CM // Computing Cluster Head and Cluster member
3. 1.Initialize the required PSO parameters
4.  $E_0 = 200$ ; // Initial energy

```

5.  $W = 0.9$  // Inertia weight
6.  $C_{rad} = 30$  // Cluster radius
7.  $S_{ra} = 36$  // sensing range 8.  $w_1, w_2, v_1, v_2$ 
   → initialize randomly
9. Node position =  $Nor_{node}$ 
10.  $P_{best} = Node\ position$ 
11.  $P_{prev\_best} = P_{best}$ 
12. Utility
   =  $[v_1\ v_2]$  //initial  $M_n$ 
   id
   (k) = lim
    $i \rightarrow 1$  to NP
   lim
    $k \rightarrow 1$  to NP
   { if  $D_i(k) < S_{range}$  0 else

13.  $P_1$  → initialize the  $i^{th}$  position
14.  $V_1$  → initialize the  $i^{th}$  velocity
15. // fitness function
16.  $\alpha_1 = 0.5; \alpha_2 = 0.2; \alpha_3 = 1 - \alpha_1 - \alpha_2;$ 

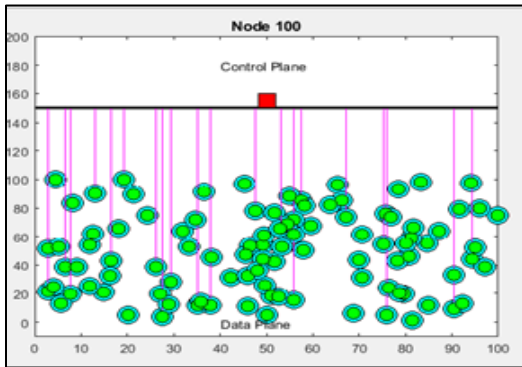
17. Where  $D(k) = \sqrt{(P_1 - Node\ position)^2}$  NP → Node position
18. If  $Min_{id}$  index is zero discard that particular index
19.  $C_n \rightarrow$  Calculate the  $Min_{id}$  index count
20.  $Fac_1$ 
   =  $\frac{1}{\sum Dist}$  // factor-1 depending upon distance between the node
    $C_n$ 

21.  $Fac_2$ 
   =  $\frac{(E_0 * ) G_n}{E_0}$ 
   //factor 2 depending upon the energy
22.  $Fac_3 = arg(Min_{id})$  //factor 3 depending upon the arguments count of  $Min_{id}$ 
23. // finding best position by using above 3 factor
24.  $B_{prebest} = \lim_{i \rightarrow 1 \text{ to } NP} \alpha_1 Fac_1 i + \alpha_2 Fac_2 i + \alpha_3 Fac_3 i$ 
25.  $C_{prev\_best\_pos} = B_{prebest}$ ;
26.  $U_{pbest\_pos} \rightarrow$  finding maximum position from  $P_{pre\_best\_pos}$ 
27.  $Utility = W Utility + w_1(P_{prev\_best} - P_{best}) + w_2(P_{prev\_best} - P_{best})$ 
28. // calculate the velocity of particle with position
29.  $P_{best} = st + Utility$  // update the position with updated velocity
30. While ( $p <$  number of nodes) 31.
     $p = p + 1;$ 
32. Find the  $B_{prebest\_pos}$ 
33. If ( $B_{prebest\_pos} > C_{prevbest}$ )
34.  $C_{prev\_best\_pos} = B_{prebest\_pos}$ 
35.  $P_{prev\_best} = P_{best}$ 
36. end if
37.  $C_{prev\_best\_pos} = B_{prebest}$ ;
38.  $U_{pbest\_pos} \rightarrow$  Finding maximum position from  $P_{pre\_best\_pos}$ 
39.  $Utility = W Utility + w_1(P_{prev\_best} - P_{best}) + w_2(P_{prev\_best} - P_{best})$ 
40. While end // repeat the process for all node, here we finding cluster head
41. Save the cluster head position

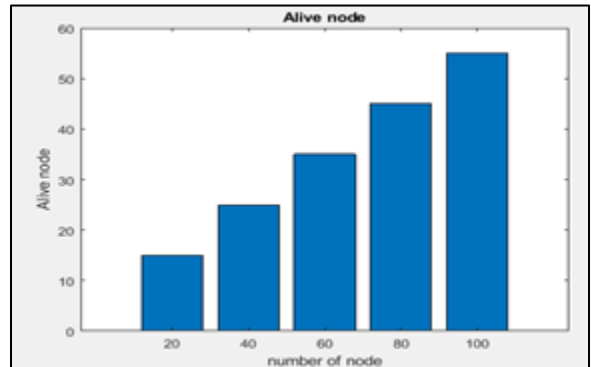
```

**V. SIMULATION AND ANALYSIS**

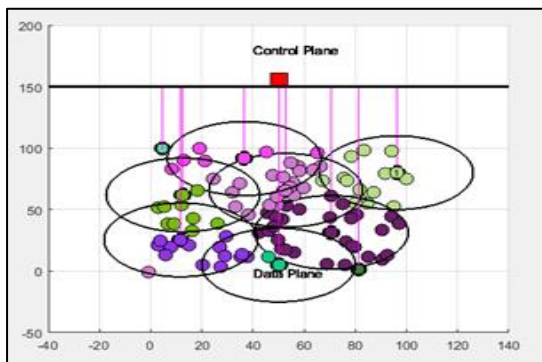
In this paper, we use MATLAB to simulate the algorithm, and compare it with traditional particle swarm optimization. In the simulation, the nodes are deployed in the area of 100m x 100m, which is also the searching area of the particle swarm optimization algorithm. Figure 1. Illustrates that 100 nodes are deployed in X and Y coordinates which is divided into data plane and control plane. Fig.2 illustrates that the cluster are formed using algorithm 1.



**Fig.1.** Nodes deployed in data plane

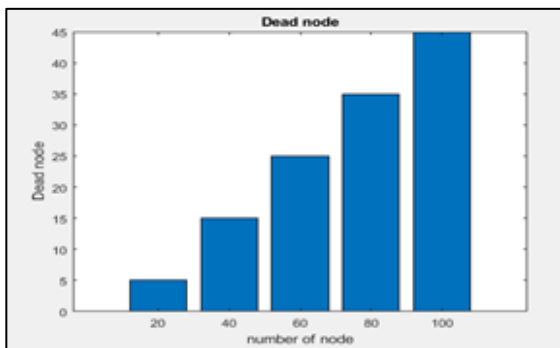


**Fig.4** Number of nodes vs Alive node



**Fig.2** Cluster formation

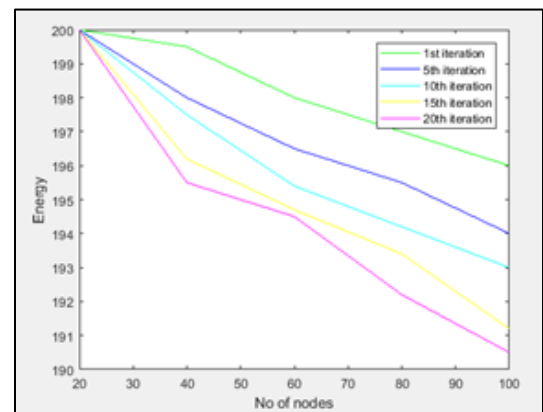
Figure.3 shows the Dead nodes in the Network with respect to the number of nodes. Initially there are 100 nodes. The above figure shows the death node count with respect to the number of nodes increased using Particle Swarm Optimization (PSO).



**Fig.3** Number of Nodes vs Dead Node

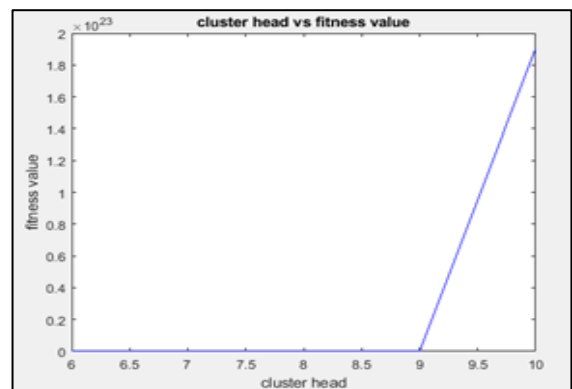
Figure.4 shows the Alive nodes in the Network with respect to the number of nodes. Initially there are 100 nodes. The above figure shows the alive node count with respect to the number of nodes increased using Particle Swarm Optimization (PSO).

Figure 5. shows the Residual Energy of the Network with respect to the number of nodes in iteration wise. The Initial Energy of the nodes are 0.5J and there are 100 nodes. It is observed that nodes life time was increased by using the Particle Swarm Optimization Algorithm.



**Fig.5** No. of nodes vs Energy

Figure 6. shows cluster head versus fitness value according to the modified PSO fitness function.



**Fig.6** Cluster head vs Fitness value

## VI. CONCLUSION

In this paper, we use non linear programming for cluster formation in SDN network is described. Then an energy efficient clustering scheme for WSN with PSO approach is presented. In the presented algorithm, the energy consumption of CHs is significantly balanced to improve the network lifetime. The presented algorithm is based on derivation of efficient particle encoding method and fitness function derivation. We have carried out thorough simulation. Simulation results clearly states that proposed PSO algorithm excels the existing algorithms.

## VII. REFERENCES

- [1] Chatterjee A., Siarry P. (2016). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers and Operations Research*, 33(3), 859-871.
- [2] Chen M.Y., Zhang C.Y., Luo C.Y. (2015). Adaptive evolutionary multi-objective particle swarm optimization algorithm, *Control and Decision*, 24(12), 1851-1855.
- [3] J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.
- [4] M. Jacqueline and C. Richard, *Application of particle swarm to multiobjective optimization*, Auburn: Auburn University, 1999.
- [5] Alfi and H. Modares, "System identification and control using adaptive particle swarm optimization," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1210–1221, 2011.
- [6] L. Li and Y. Du, "Application of Modified Particle Swarm Optimization in Node Locating of Wireless Sensors Networks", *Computer Applications and Software*, vol. 31, no. 4, (2014), pp. 69-72.
- [7] Y. Wang and J. Yang, "Localization in wireless sensor network based on improved particle swarm optimization algorithm", *Computer Engineering and Applications*, vol. 50, no. 18, (2014), pp. 99-102.
- [8] Zungeru, Murtala A, Ang L, Seng KP. Classical and swarm based routing protocols for wireless sensor networks: a survey and comparison. *Journal of Network and Computer Applications*. 2012 Sep; 5(35):1508–36.
- [9] Sundaran K, Ganapathy V. Energy efficient wireless sensor networks using dual cluster head with sleep/active mechanism. *Indian Journal of Science and Technology*. 2016 Nov; 9(41):1–6.
- [10] Joseph PS, Balaj CD. Transmission loss minimization using optimization technique based on PSO. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*. 2013 May – Jun; 6(1):1–5.