

Simple Clustering for Wireless Sensor Networks

R. Rathna

*Department of Information Technology, Sathyabama University, Chennai, Tamilnadu, India
Email: rathna0809@gmail.com*

Abstract: *Traditional Wireless Sensor Networks (WSNs) are designed for connecting pure electronic or mechanical sensors through wireless communication. Nowadays, WSNs are combined with other technologies like big data and cloud for data management. While combining the sensors (low level devices) with a high end network like Internet, data communication should be taken care of in the gateway between these two networks. With this type of WSNs, if all the sensors are in direct communication with the high end network, then a lot of energy will be consumed, requiring complex protocols for making this communication possible. So, to reduce the energy wastage, the clustering concept is used in WSN. Although so many complex and innovative clustering techniques are available for WSN, this paper describes a very simple clustering technique for environmental monitoring WSN with proven results.*

Keywords: *Clustering algorithms, data transfer, energy consumption, radiofrequency, sensor systems and applications, zigbee.*

1. Introduction

Environmental monitoring is an important application of Wireless Sensor Network (WSN). Acquisition of precise data and immediate transfer of the data to the sink node in time is very crucial for environmental monitoring applications. From the survey of several WSN applications, it is clear that the energy consumption of any type of sensor network depends mostly on the data communication between the nodes. Nodes from different types of available vendors are nowadays designed to have sleep mode, so that, if they do not have anything to send or receive, the radio can enter a sleep state. However, the nodes can communicate only if they are active.

Time Division Multiple Access (TDMA) based schedules assign unique time slots for each node to send and receive the data. The main advantage of this type of protocol

is that it will avoid interference between adjacent wireless links. So, there will be less energy wastage due to packet collision. Also, TDMA can solve the hidden terminal problem without additional message overhead because neighbouring nodes transmit at different time slots (Li Deliang and Peng Fei [1]).

When compared with IEEE 802.11, TDMA-based algorithms prolong the lifetime of WSN, even when used for a greater number of nodes. Every node will be given a separate time slot for sending and receiving the data. But the schedules should be created in such a way, that the nodes should not interfere while sending or receiving. There are two types of TDMA Scheduling – one hop TDMA scheduling and multi-hop TDMA scheduling (Sinem Coleri Ergen and Pravin Varaiya [2]). If there is a low number of nodes, then all the nodes can be placed around the sink node or Base Station (BS). In that case we can go for one hop TDMA scheduling. If a higher number of nodes is used in a wide area of deployment, then all the nodes cannot reach the sink node in a single hop. For the environmental monitoring applications, a greater number of nodes are deployed in a wide spread area. So, the multi-hop TDMA scheduling is the only choice.

All the self-organizing MAC protocols like Aloha with PS, B-MAC (Berkeley Medium Access Control [3]), S-MAC (MAC protocol designed for WSN [4]), X-MAC (a short preamble MAC protocol by Buettner et al. [5]), C-MAC (MAC layer protocol using Convergent Packet Forwarding by Liu et al. [6]), MX-MAC (a small modification of X-MAC, where the data itself is sent repeatedly until it gets an acknowledgement from the receiver), Mix-MAC (chooses the efficient schedule from a group of compatible MAC schedules), WiseMAC (designed for Infrastructure Wireless Sensor Network by El-Hoiydi and Decotignie [7]), SpeckMAC[8], Z-MAC (Zebra MAC by Rhee et al. [9]) and S-MAC [10] are based either on Preamble sampling or adaptable duty cycle based or by using synchronization information. The TDMA based protocols like LEACH, DECSA (not purely TDMA [11]), HM-VMPC [12], MC-LMAC [13] and EA-TDMA [14] are slot-based and there, by interference, can be avoided. Among them, some are application specific and some are complex. For the environmental monitoring applications, specific protocols are only suitable, since they depend on the external factors.

2. Environmental monitoring WSN

For environmental monitoring applications, the number of nodes deployed in a region is always very high. The objective of such nodes is to observe the environment by tracking any physical parameter according to the sensors used for that application (Farahani [15]). All the nodes are equipped with sensors in an environmental monitoring WSN. Because in most cases, all the predesigned nodes are deployed in the site randomly by throwing them from any aerial vehicle, since some places of monitoring are very remote with extreme climate conditions.

2.1. WSN for environmental impact assessment

WSNs are used for variety of environmental monitoring applications. There are WSNs to monitor the Environmental Impact Assessment (EIA). Valverde et al. [16] has described about a WSN to assess the impact of the processes taking place in and around a coffee factory. In a coffee factory a lot of waste water goes out on a daily basis. Not only waste water, along with that lot of poisonous gas also goes out. Measures are always there for all such factories to check their danger level, if all the process in the manufacturing unit works correctly. But, if anything goes wrong by mistake, the waste water and gases produced as a result of such an event will have a disastrous effect on the surroundings. The WSN developed in this work deploys many sensors for monitoring the PH level of the waste water, humidity, temperature and certain dangerous gases in the surroundings. Five different types of nodes were placed in five different places in the coffee factory and the sink node was kept inside the main building. The sink node creates the network after receiving the identification messages from all the nodes. The data, collected from all these nodes, is aggregated in the sink node. Following this, the application was tested by two different arrangements in the deployment. 'Cookie' is the sensor hardware used in this application. The corresponding software for interfacing this hardware is used on the server side. This application clearly describes the real time deployment of a WSN in the field of environmental monitoring. It is very much useful for the food factories in European countries for the EIA purpose.

2.2. Usage of WSN in smart grid

The WSN is used for efficient electric power utilization in smart grids. Gungor, Lu and Hancke [17] explains the WSN usage in smart grid monitoring. The basic need for proper electricity management is a must these days. Due to improper usage of electricity, sudden power cuts have been happening frequently in recent days. The smart grid is a boon to this problem. It monitors the power utilization in a building or a specific area. Online sensing technologies are generally used. Traditional systems use wired monitoring systems, which have become unreliable now. The work explains WSN for monitoring the electrical devices. It has been implemented in the power generation, delivery and monitoring units. TmoteSky motes are used as sensor nodes in this application. The impact of different parameters, responsible for assessing electricity utilization, is measured in the WSN, deployed in different types of power-electrical environments. The final report gives the status of the electricity management in the taken environment. In general, for all the environmental monitoring applications, there will be many sensor nodes and they are always densely deployed. In such type of applications, the inter distance between the nodes is always shorter. For such short communication range, most of the sensor applications use zigbee standard radio [18].

All the monitoring applications use WSN in dense deployment and some even with RFID [19]. In such a densely deployed WSN, if all the nodes, after sensing the data try to find a route to reach the sink by passing it to a nearest neighbour, then there will be increased traffic leading to more collisions. All these factors will deplete the energy of the WSN. The solution to this is to go for clustering.

3. Clustering in WSN

Clustering is applied in all domains. It is applied for data in data mining and retrieval techniques [20] and for nodes in the WSN. Generally, all the nodes in the initial stage decide to become the Cluster Head and they will start advertising to their k -hop neighbours. Then the level 1 CHs will try to become level 2 CHs with some probability value greater than other nodes in the network. This will continue until all the nodes are categorized under some clusters. This is the basic methodology followed in Hierarchical Clustering (Huang, Wang and Chen [21]). In this algorithm Cluster Based Energy Efficient Protocol (CBERP), the LEACH was enhanced by increasing the number of nodes in the set of advanced nodes- which are eligible for CH selection. By doing this way, the life time of advanced nodes can be increased.

In general, a hierarchical clustering starts by grouping nodes first into a tree. This tree is considered to be made up of clusters. Hierarchical clustering is classified into two types. They are agglomerative and divisive. In the first type, the clustering is done from bottom-up by decomposing. In the second type, the clustering is done by top-down. The bottom up approach is basically a merging technique, in which, the individual clusters unite with other nearby clusters forming a big single cluster covering all the nodes in the WSN. The second one is just the contrary of the bottom-up approach. In this technique, the entire network is initially considered as a single cluster. Then, it splits the big cluster into several small clusters based on some criteria. His top- down hierarchical clustering is the approach followed here for initial clustering.

“D-hop” clusters are created with the maximum run time of $O(d)$ rounds by the Max-Min d -cluster Algorithm [22]. In this approach the nodes which are at a distance of d or less than that from the cluster head are clustered together under the same cluster. It happens by following two rounds. Flooding is done during the two rounds. Cluster IDs or numbers are recorded after every round. All the nodes send their IDs to all the neighbour nodes of their reach. This is called flooding. Then the nodes receiving more numbers of the same node id follow some conditions to select their CH. In the second round, for the smallest node IDs, if any node after flooding receives its own ID, then it declares itself as the CH. Otherwise, the node ID received by most of the other nodes becomes the CH as in first round. To select the gateway node, every node has to broadcast its CH. If a node receives different CH IDs from all its neighbour nodes, then that node sends its ID as the gateway node to all the CHs through other nodes.

Although this algorithm creates the clusters from n number of nodes without leaving behind any individual nodes, it cannot be called an energy efficient clustering. Also, it has been carried out with an assumption that every node in the network has the knowledge about the full network topology. For large number of nodes, this assumption is impossible to take into account. The nodes randomly placed in a location are to be clustered to reduce the excess network traffic, as given in the initial part of this work. The complex clustering techniques in other fields should not be used here. Hence, a new clustering technique is used here based on the above said two algorithms.

3.1. 2-Tier hierarchical distance based clustering

Hierarchically, the cluster formation is done here. In a loop, a node (randomly) is taken as the initial node. From that node those nodes which are of one hop and two hop distance away are treated as in a cluster. The first node which comes out of this condition is taken as the initial node for second cluster.

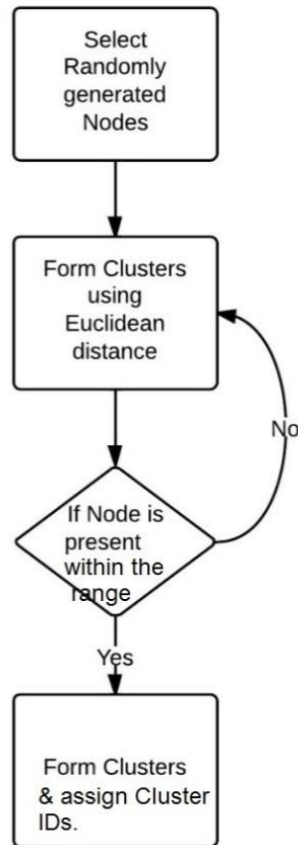


Fig. 1. Clustering

The clusters are formed based on the Euclidean distance. The sensor nodes within the small distance are grouped together and form clusters. The Euclidean distance can be calculated using the following formula (1).

$$(1) \quad d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

The maximum distance between the sensor nodes in the cluster is 25 m. Based on the first sensor node, calculate the distance with other sensor nodes and then those nodes which satisfy the distance condition are formed as a cluster. This is done iteratively by starting randomly from a node and it stops, once all the nodes in the Network are clustered. This is implemented in MATLAB.

Fig. 2 explains the process.

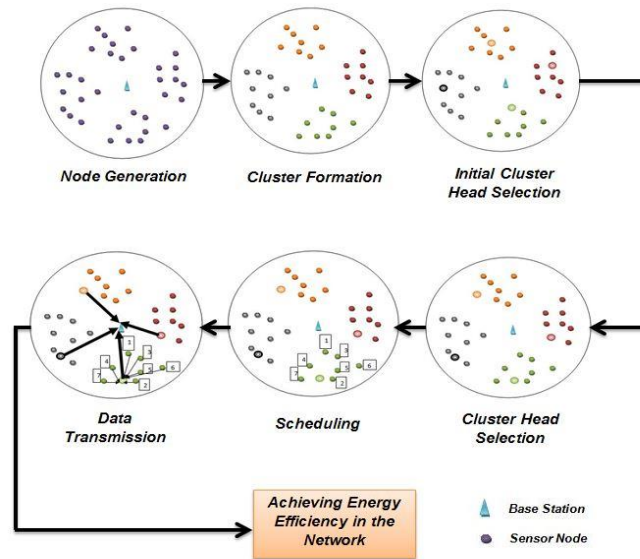


Fig. 2. Architecture of the proposed method

3.2. Algorithm

The clustering covers the entire number of nodes in the network. So after the loop finishes, all the nodes will be in a cluster. The simulation is done for 100 nodes randomly placed in a 100×100 m area in Matlab forming six clusters.

A flag parameter which is of Boolean type is assigned for every node. By default, the parameter value is “False” for all nodes. Once it gets assigned in a cluster, that node’s flag is marked “True”.

The pseudocode for the clustering algorithm is:

```

for j=1
for i=1:1:n
    dis=((S(i).xd) – Xk)^2 + ((S(i).yd) – Yk)^2;
    d=sqrt(dis);
    if(d<25 && (S(i).flag==false))
        S(i).C=1;
        S(i).flag=true;
        Plot (S(i).xd, S(i).yd, 'm.');
```

Node_Id_in_Cluster1=S(i).id

```

    end
end
for i=1:1:n
    if(S(i).flag==false)
        Xk=S(i).xd;
        Yk=S(i).yd;
    end
end
end
end
```

Here two loops are used. The first loop is for the cluster iteration. The second loop is for the iteration of nodes in that cluster; $S(i)$ denotes the node ID; C represents the cluster ID; X_k and Y_k represents the location of the first node.

The distance from the first node (random node taken as first node) is calculated using the Euclidean distance formula to neighbor node. If that distance is less than 25 m and the neighbor node is having its flag parameter “false”, then that node is added to the first node forming a cluster.

Then the first node which does not met the condition gets the flag value “false”. Its location is assigned for X_k and Y_k . It acts as the first node for the second cluster. It continues till the flags of all the nodes in the network get the value of “true”. Several iterations are done using this algorithm. As a result of this simulation, six clusters are formed with uneven number of nodes in each cluster, during the first iteration. Table 4.2 shows the nodes and their corresponding clusters out of the first iteration. The Hierarchical clustering procedure starts with the node ID 1 for the first iteration. After the cluster formation, a random value is assigned as the node weighting parameter. Based on this weight the CH is selected initially for the first round. Then, from the next round onwards the nodes are weighted based on their load L and residual energy E . Using the following equation (2) the weight α is calculated (where i denotes the node ID and t denotes the time instant),

$$(2) \quad \alpha = KE_{it} + \frac{1}{L_{it}}.$$

According to the increasing order of α , the schedule is created for every cluster. The nodes in the cluster, if they have any load to send, use the slots according to that schedule for data transmission. If a node inside that cluster has anything to send, its energy is checked first. If it has positive energy, then according to its α value, it gets a slot. In that slot, it forwards its load (packet) to the node which has the next higher α value in the schedule. At the end of every round, the Cluster Head (CH) of all the clusters has the data transferred from the nodes inside that cluster. The node which gets the last slot in the schedule is the Cluster Head. This CH has the maximum α value. The data from the node reaches the sink node by following this schedule inside the cluster. The data from CH forwards it to the nearby node in the neighbor cluster, which in turn forwards the data to its CH. Using this incremental forwarding, the data finally reaches the Sink node (or) Base Station.

4. Results

The hierarchical clustering described here starts randomly from a node and it proceeds further clustering all the nodes in the WSN. The execution stops once all the nodes are classified in clusters. The algorithm is implemented for three iterations starting from node ID 1, 28 and 76 (Tables 1, 2 and 3).

Table 1. Clustered nodes during the first iteration

Cluster No	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Nodes in cluster	1	11	2	6	12	21
	3	20	8	43	18	26
	7	25	10	98	23	29
	13	34	14		27	31
	17	37	15		38	33
	28	46	16		61	40
	32	49	19		67	52
	39	50	22		81	56
	45	51	41		82	66
	57	54	44		91	68
	62	55	47		94	71
	86	58	48			76
	97	60	65			84
		63	70			93
		64	74			
		77	78			
		85	99			
		87				
		92				
		95				
	96					
	100					

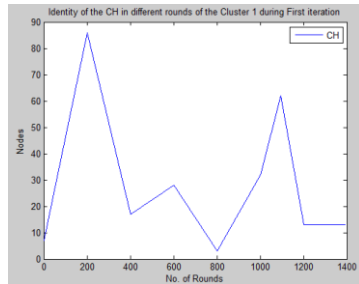
Table 2. Clustered nodes during the second iteration

Cluster No	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Nodes in cluster	1	2	3	4	5	9
	7	6	17	15	21	18
	13	8	39	22	29	20
	24	10	57	41	30	34
	28	14	86	70	33	38
	32	16	97	72	40	42
	35	19		75	52	55
	36	37		79	56	85
	45	43			80	90
	53	44			88	96
	59	46				
	62	47				
	69	48				
	73	54				
	83	64				
	89	77				
		78				
		95				
		98				
	99					

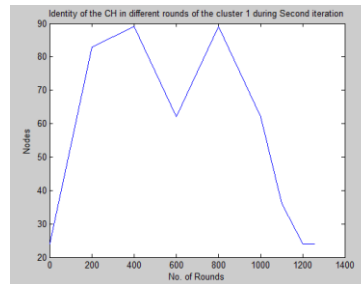
Table 3. Clustered nodes during the third iteration

Cluster No	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Nodes in cluster	1	2	3	4	5	8
	6	23	16	15	7	17
	12	27	24	44	9	25
	18	31	47	62	11	30
	36	53	58	69	14	33
	37	54	64	84	19	35
	42	57	95	98	20	45
	48	79		100	34	49
	50	82			38	61
	51	85			40	68
	52	86			41	70
	55	87			43	72
	60	93			46	75
	76	99			59	80
	78				63	
	90				67	
	91				81	
					83	
				89		
				92		

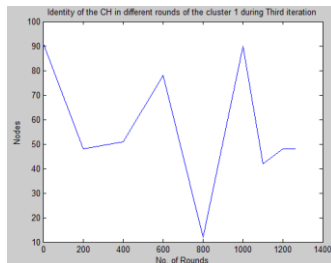
The selection of CH is based on the node weight parameter α . So the position of CH dynamically changes for every run. This will prevent fast energy drain in any single node, because of being the CH. Initially the nodes are arranged by random distribution in the given area. So there is a chance that some nodes may get distributed too far away from any node in the WSN. Because of this they may not get clustered in any of the clusters.



(a) CH=7, 86, 17, 28, 3, 32, 62, 13, 13, 13



(b) CH=24, 83, 89, 62, 89, 62, 36, 24, 24



(c) CH=91, 48, 51, 78, 12, 90, 42, 48, 48

Fig. 3. Dynamically changing behaviour of CH in cluster 1 during the first (a), second (b) and third (c) iterations

In all the three iterations, it has been observed that during CH selection, only the node which has maximum energy and at the same time minimum load only can become a CH. Some nodes are there which have not been selected as the CH at all during the full 1400 rounds. And some nodes have obtained CH status more than one time also. The overall objective is to reduce the energy consumption during the full set of rounds and not like every node should become CH at least once.

The following graphs show the dynamic behaviour of CH selection. For showing the dynamic nature of CH, the cluster 1 is taken as sample and the graphs are drawn for the three iterations. The graph (Fig. 3a) shows that the nodes 7, 86, 17, 28, 3, 32, 62 and 13 are getting the CH position during the first iteration. The graph (Fig. 3b), shows that the nodes 24, 83, 89, 62, 89 and 36 are getting the CH position during the second iteration. The graph (Fig. 3c), shows that the nodes 91, 48, 51, 78, 12, 90 and 42 are getting the CH position during the third iteration. Only two or three nodes are getting the CH position repeatedly.

The initial energy of any node in this study is 0.5 J. During data transmission and transfer of control packets its energy reduces. Apart from that gradually a very meagre amount of energy leaks without any processing or radio usage. The energy which is consumed during the usage of radio alone is taken for residual energy calculation. To analyse the residual node energy of all the nodes, the results from second iteration where clustering starts from node ID 28 are taken for observation (Table 4).

Table 4.1. Second iteration-residual energy of nodes during the rounds 1, 200, 400, 600, 800, 1000, 1103, 1200 and 1261 (Cluster 1)




Cluster ID	Node ID	Residual energy of nodes during the rounds									
		Initial energy	1	200	400	600	800	1000	1103	1200	1261
Cluster 1	1	0.5	0.4998	0.4203	0.3339	0.2476	0.1579	0.0716	0.0244	0	0
	7	0.5	0.4998	0.4145	0.3322	0.2467	0.1645	0.0789	0.0372	0	0
	13	0.5	0.4998	0.4212	0.3391	0.2537	0.1715	0.0861	0.0443	0.0007	0
	24	0.5	0.4965	0.4085	0.3201	0.2352	0.1469	0.062	0.0155	0	0
	28	0.5	0.4998	0.408	0.3258	0.2404	0.1583	0.0729	0.0312	0	0
	32	0.5	0.4998	0.4114	0.3261	0.2376	0.1523	0.0638	0.0205	0	0
	35	0.5	0.4997	0.4082	0.3196	0.2344	0.1458	0.0606	0.0139	0	0
	36	0.5	0.4997	0.4159	0.3285	0.2444	0.1568	0.0727	0.0266	0	0
	45	0.5	0.4998	0.4178	0.3322	0.2499	0.1643	0.0821	0.037	0	0
	53	0.5	0.4998	0.4132	0.3262	0.2426	0.1556	0.072	0.0262	0	0
	59	0.5	0.4998	0.4106	0.3244	0.2383	0.1554	0.0693	0.0272	0	0
	62	0.5	0.4997	0.4173	0.3279	0.2386	0.1525	0.0632	0.0194	0	0
	69	0.5	0.4998	0.4206	0.3344	0.2483	0.1589	0.0727	0.0257	0	0
	73	0.5	0.4998	0.4095	0.3256	0.2383	0.1543	0.067	0.0243	0	0
	83	0.5	0.4998	0.4142	0.3316	0.2457	0.163	0.0771	0.0352	0	0
	89	0.5	0.4998	0.4192	0.3317	0.2474	0.1598	0.0755	0.0327	0	0

Table 4.2. Second iteration-residual energy of nodes during the rounds 1, 200, 400, 600, 800, 1000, 1103, 1200 and 1261 (Clusters 2-6)

Cluster ID	Node ID	Residual energy of nodes during the rounds									
		Initial energy	1	200	400	600	800	1000	1103	1200	1261
Cluster 2	2	0.5	0.4998	0.4156	0.3314	0.2513	0.1671	0.0828	0.0359	0	0
	6	0.5	0.4998	0.423	0.3376	0.2522	0.1669	0.0815	0.0382	0	0
	8	0.5	0.4998	0.4077	0.3239	0.2401	0.1563	0.0725	0.0299	0	0
	10	0.5	0.4956	0.4059	0.32	0.2343	0.1485	0.0627	0.0191	0	0
	14	0.5	0.4998	0.4161	0.3323	0.2485	0.1647	0.0809	0.0383	0	0
	16	0.5	0.4998	0.4189	0.3335	0.2482	0.1628	0.0775	0.0342	0	0
	19	0.5	0.4998	0.4134	0.3267	0.2359	0.1492	0.0626	0.0186	0	0
	37	0.5	0.4997	0.4063	0.3129	0.2236	0.1344	0.0451	0	0	0
	43	0.5	0.4998	0.4147	0.3253	0.2401	0.1549	0.0697	0.0265	0	0
	44	0.5	0.4998	0.407	0.3184	0.2298	0.1413	0.0527	0.0035	0	0
	46	0.5	0.4998	0.4099	0.3241	0.2382	0.1523	0.0665	0.0187	0	0
	47	0.5	0.4998	0.4202	0.3362	0.2522	0.1683	0.0843	0.0417	0.0004	0
	48	0.5	0.4998	0.4181	0.3362	0.2501	0.1641	0.078	0.0343	0	0
	54	0.5	0.4998	0.4092	0.3228	0.2362	0.1497	0.0632	0.0193	0	0
	64	0.5	0.4998	0.4188	0.3335	0.2482	0.1629	0.0776	0.0343	0	0
	77	0.5	0.4998	0.4229	0.3375	0.252	0.1666	0.0811	0.0378	0	0
	78	0.5	0.4998	0.4156	0.3312	0.2468	0.1625	0.0781	0.0353	0	0
95	0.5	0.4998	0.4157	0.3356	0.2513	0.167	0.0827	0.0399	0	0	
98	0.5	0.4998	0.4239	0.3395	0.2551	0.1707	0.0863	0.0435	0.0019	0	
99	0.5	0.4998	0.4047	0.3176	0.2307	0.1437	0.0568	0.0126	0	0	
Cluster 3	3	0.5	0.4998	0.4316	0.3499	0.2695	0.189	0.1073	0.0658	0.0208	0
	17	0.5	0.4998	0.4123	0.3328	0.2534	0.1728	0.0933	0.0524	0.0139	0
	39	0.5	0.4998	0.4206	0.3412	0.2606	0.1813	0.1019	0.061	0.0213	0
	57	0.5	0.4998	0.4118	0.3307	0.2509	0.171	0.0899	0.0488	0.0101	0
	86	0.5	0.4998	0.426	0.3461	0.265	0.185	0.1051	0.0627	0.024	0
	97	0.5	0.4986	0.4169	0.3372	0.2575	0.1766	0.0969	0.0559	0.0173	0
Cluster 4	4	0.5	0.4998	0.4119	0.33	0.2482	0.1663	0.0845	0.0422	0	0
	15	0.5	0.4998	0.4119	0.3285	0.245	0.1615	0.078	0.0348	0	0
	22	0.5	0.4997	0.4219	0.3372	0.2525	0.1679	0.0832	0.0394	0	0
	41	0.5	0.4998	0.4151	0.3317	0.2483	0.1649	0.0814	0.0398	0	0
	70	0.5	0.4998	0.4122	0.3289	0.2457	0.1624	0.0792	0.0361	0	0
	72	0.5	0.4998	0.4264	0.3434	0.2603	0.1773	0.0943	0.0514	0.0098	0
	75	0.5	0.4998	0.4145	0.3277	0.2408	0.1539	0.067	0.022	0	0
79	0.5	0.4982	0.4153	0.3308	0.2462	0.1616	0.0771	0.0333	0	0	
Cluster 5	5	0.5	0.4998	0.4266	0.3471	0.2656	0.1841	0.1026	0.0592	0.0191	0
	21	0.5	0.4998	0.4185	0.337	0.2555	0.174	0.0925	0.0511	0.011	0
	29	0.5	0.4998	0.423	0.34	0.257	0.174	0.091	0.0488	0.008	0
	30	0.5	0.4998	0.4134	0.3288	0.2441	0.1595	0.0749	0.0319	0	0
	33	0.5	0.4998	0.4232	0.3423	0.2595	0.1766	0.0937	0.0517	0.0109	0
	40	0.5	0.4998	0.4106	0.3273	0.246	0.1646	0.0833	0.042	0.002	0
	52	0.5	0.4978	0.4185	0.337	0.2555	0.174	0.0925	0.0511	0.011	0
	56	0.5	0.4998	0.4123	0.3286	0.247	0.1653	0.0837	0.0422	0.002	0
	80	0.5	0.4998	0.4102	0.3283	0.2464	0.1645	0.0827	0.0391	0	0
88	0.5	0.4998	0.4219	0.3396	0.2574	0.1751	0.0929	0.0491	0.0107	0	

Table 4.2. (continued)

Cluster ID	Node ID	Residual energy of nodes during the rounds									
		Initial energy	1	200	400	600	800	1000	1103	1200	1261
Cluster 6	9	0.5	0.4998	0.4264	0.3426	0.2589	0.1752	0.0915	0.049	0.0078	0
	18	0.5	0.4998	0.4163	0.3325	0.2507	0.169	0.0872	0.0457	0.0054	0
	20	0.5	0.4998	0.4208	0.3395	0.2581	0.1768	0.0954	0.0542	0.0141	0
	34	0.5	0.4998	0.4126	0.3311	0.2495	0.168	0.0865	0.0451	0.005	0
	38	0.5	0.4998	0.4107	0.3294	0.248	0.1667	0.0853	0.0441	0.004	0
	42	0.5	0.4998	0.4291	0.3459	0.2627	0.1795	0.0964	0.0541	0.0132	0
	55	0.5	0.4998	0.4188	0.3395	0.2582	0.1769	0.0957	0.0524	0.0144	0
	85	0.5	0.4998	0.415	0.332	0.2489	0.1659	0.0829	0.0407	0	0
	90	0.5	0.4978	0.414	0.332	0.2501	0.1682	0.0862	0.0426	0.0043	0
	96	0.5	0.4998	0.4164	0.3346	0.2528	0.171	0.0893	0.0457	0.0075	0

-  – Cluster head
-  – Dead node
-  – Cluster head and dead node in N-th round

In the Cluster1, the node 36 even though it is a CH, has not been depleted of its energy during the round 1103. One node (13) has survived even in round 1200. In Cluster 2, the node 46 which is a CH survives even in round 1103 and two nodes (47 and 98) survive in round 1200. In Cluster 3, all the nodes survive in round 1200. This is so because, only six nodes are there inside that cluster. In Cluster 4, the node (4) which is the CH in round 1200 dies first. But that also happens only in the round 1200. In Cluster 5, out of the ten nodes only two nodes (30 and 80) die in round 1200. And also the CH survive in round 1200. In Cluster 6, the node which is the CH alone dies in round 1200. From the data above, it has been observed that the nodes started to die at the 1103rd round. During the round 1261 all the nodes are dead. The sustainability of the nodes is improved by using this algorithm. The overall energy of the WSN, at the end of certain rounds are also compared. That tabulation for the first, second and third iterations are given below in Tables 5, 6 and 7.

Table 5. Remaining energy (J) of the WSN during specific rounds – first iteration

Rounds	Before 1st round	1	200	400	600	800	1000	1096	1200	1300	1394
Remaining energy	40	39.97	33.17	26.34	19.51	12.67	5.84	2.59	0.23	0.09	0

Table 6. Remaining energy (J) of the WSN during specific rounds – second iteration

Rounds	Before 1st round	1	200	400	600	800	1000	1103	1200	1261
Remaining energy	35	34.97	29.12	23.25	17.37	11.49	5.62	2.59	0.27	0

Table 7. Remaining energy (J) of the WSN during specific rounds – third iteration

Rounds	Before 1st round	1	200	400	600	800	1000	1103	1200	1261
Remaining energy	40	39.97	33.23	26.47	19.7	12.933	6.17	2.89	0.21	0

The comparison of important features from all the three iterations are made and are tabulated down in Table 8.

Table 8. Comparison of the three iterations

Features	Iteration 1	Iteration 2	Iteration 3
Nodes getting clustered	80	70	80
No of nodes in Cluster 1	13	16	17
No of nodes in Cluster 2	22	20	14
No of nodes in Cluster 3	17	6	7
No of nodes in Cluster 4	3	8	8
No of nodes in Cluster 5	11	10	20
No of nodes in Cluster 6	14	10	14
Round in which first node dies	1096	1103	1103
Round in which all the nodes died	1394	1261	1266

At the end of all the three iterations, it has been observed that the first node dies around the round 1100. The round, in which all the node dies, is also compared and presented in Table VIII.

Even though the first node dies around the round 1100, the other nodes continue the LECSA by removing the dead node from the list. This increases the longevity of the network. Table 9 shows the numbers of living nodes in six clusters during the three iterations in round 1200.

Table 9. Number of living nodes in round 1200 during the iterations 1, 2 and 3

Cluster ID	Iteration 1	Iteration 2	Iteration 3
Cluster 1	1	1	0
Cluster 2	0	1	2
Cluster 3	1	6	7
Cluster 4	3	1	8
Cluster 5	2	8	1
Cluster 6	2	9	2

From the comparison (Table 9), it is clear that even after the first node dies around the round 1100, the WSN continues to work with most of the clusters with not much degradation for 100 more rounds. This proves that this algorithm works for all WSNs and Wireless Mesh Networks [23], better than other existing protocols [24]. One more comparison is done to measure how this algorithm proves better over the other existing techniques.

The percentage of energy loss for the proposed LECSA is compared with the existing protocols, namely LEACH, the multi-hop variant of LEACH called as MR-LEACH [25], a cluster-based routing protocol called ACT [26], an Energy Efficient Clustering Algorithm (EECA) [27] and a Separating Cluster-based Algorithm-SCA [28]. The results are shown in the Table 10.

Table 10. Comparison of energy loss

S. No	Protocol	Percentage of energy loss at the end of round 600
1	LEACH	69.23
2	MR-LEACH	64.2
3	EECA	56
4	ACT	50
5	SCA	61.11
6	LECSA	53.04

From Table 10, it is clear that the proposed LECSA works with better energy efficiency than that of the LEACH, MR-LEACH, SCA where energy balance is done among the sensors in each cluster and EECA where the tree aggregation is done based on assigning weight for the CHs. ACT shows slightly less energy loss because of the difference in cluster size around the BS. But changing the cluster radius based on the node location in-order to assign equal energy for all clusters is complex.

5. Conclusion

The overall energy consumption of the WSN is reduced when this model is implemented. Several iterations are done for the clustering and the following scheduling and their results are presented in this paper. Most of the environmental monitoring works, in particular the pollution monitoring, talks about several new types of clustering techniques and scheduling techniques in simulation. But their real time implementation follows a simple topology and most probably a centralized approach of their earlier described protocol. By using this simple clustering technique, the energy of the nodes will not get depleted easily. Since the node with least energy is given priority here, the overall energy consumption of the network greatly reduces.

References

1. Li, D., P. Fei. Energy-Efficient MAC Protocols for Wireless Sensor Networks. – In: Proc. of IEEE Infocom, New York, USA, 2009, pp. 32-36.
2. Ergen, S. C., P. Varaiya. TDMA Scheduling Algorithms for Wireless Sensor Networks. – Wirel. Netw., Vol. **16**, 2010, No 4, pp. 985-997.
3. Jurdak, R., A. G. Ruzzelli, G. M. P. O'Hare. Radio Sleep Mode Optimization in Wireless Sensor Networks. – IEEE Transactions On Mobile Computing, Vol. **9**, 2010, No 7, pp. 955-968.
4. Ye, W., J. Heidemann, D. Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. USC/Information Sciences Institute, 2003.
5. Buettner, M., G. V. Yee, E. Anderson, R. Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. – In: Proc. of 4th International Conference on Embedded Networked Sensor Systems, ACM, 2006, pp. 307-320.

6. Liu, S., K. W. Fan, P. Sinha. CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks, SECON, 2007.
7. El-Hoiydi, A., J. D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. – In: Proc. of Ninth International Symposium on IEEE, Computers and Communications, ISCC'2004. Vol. 1, 2004, pp. 244-251.
8. Wong, K. J., D. K. Arvind. SpeckMAC: Low-Power Decentralised MAC Protocols for Low Data Rate Transmissions in Specknets. – In: Proc. of 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality, ACM, 2006, pp. 71-78.
9. Rhee, I., A. Warrier, M. Aia, J. Min, M. L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. – IEEE/ACM Transactions On Networking (TON), Vol. 16, 2008, No 3, pp. 511-524.
10. Ghosh, S., P. Veeraghavan, S. Singh, L. Zhang. Performance of a Wireless Sensor Network MAC Protocol with a Global Sleep Schedule. – A Transaction in International Journal of Multimedia and Ubiquitous Engineering, Vol. 4, 2009, No 2, pp. 99-113.
11. Yong, Zhu, Qing Pei. A Energy-Efficient Clustering Routing Algorithm Based on Distance and Residual Energy for Wireless Sensor Networks. – Sci. Verse Science Direct Procedia Engineering, Vol. 29, 2012, pp. 1882-1888.
12. Yang, Y., Y. Ju, H. Xia, W. Zhao, Y. Zhen. A Network Protocol Stack Based Radiation Sensor Network for Emergency System. – IJCSNS International Journal of Computer Science and Network Security, Vol. 8, 2008, No 8, pp. 312-318.
13. Incel, O. D., L. Van Hoesel, P. Jansen, P. Havinga. MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks. – An Elsevier Transaction in the Journal of "Ad Hoc Networks", Vol. 9, 2011, No 1, pp. 73-94.
14. Shafulah, G. M., A. Thompson, P. Wolfs, S. Ali. Energy-Efficient TDMA MAC Protocol for Wireless Sensor Networks Applications. – In: Proc. of 5th ICECE, Dhaka, Bangladesh, 24-27 December 2008, pp. 85-90.
15. Farahani, S. ZigBee Coexistence, in ZigBee Wireless Networks and Transceivers. Newnes, Burlington, 2008, pp. 247-259.
16. Valverde, J., V. Rosello, G. Mujica, J. Portilla, A. Uriarte, T. Riesgo. Wireless Sensor Network for Environmental Monitoring: Application in a Coffee Factory. – International Journal of Distributed Sensor Networks, 2012.
17. Gungor, V. C., B. Lu, G. P. Hancke. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. – Industrial Electronics, IEEE Transactions, Vol. 57, 2010, No 10, pp. 3557-3564.
18. Gavaskar, V., T. Sasipraba. Power Optimization in Wireless Sensor Networks Using Dynamic Scaling Algorithms. – National Journal on Electronic Sciences and Systems, Sathyabama University, Vol. 2, 2011, No 2, pp. 67-75.
19. Anu, M., V. Anandha, G. S. Mala. RFID Data Encoding Scheme in Supply Chain Management with Aid of Orthogonal Transformation and Genetic Algorithm (GA). – International Review on Computers and Software (IRECOS), Vol. 8, 2013, No 11, pp. 2562-2569.
20. Gowri, S., G. S. Anandha Mala, G. Divya. Enhancing the Digital Data Retrieval System Using Novel Techniques. – Journal of Theoretical and Applied Information Technology, 1992-8645, Vol. 66, 2014, No 2, pp. 481-489.
21. Huang, Y.-F., N.-C. Wang, M.-C. Chen. Performance of a Hierarchical Cluster-Based Wireless Sensor Network. – In: Proc. of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC'08), 2008, pp. 349-354.
22. Abbasi, A. A., M. Younis. A Survey on Clustering Algorithms for Wireless Sensor Networks. – Computer Communications, 2007, pp. 2826-2841.
23. Valarmathi, K., N. Malmurugan. Bandwidth Reservation Technique Based on Traffic Priority for Wireless Mesh Networks. – National Journal on Advances in Computing and Management, Sathyabama University, Vol. 3, 2012, No 1, pp. 8-14.
24. Leni, Sam, A. Ezil, S. K. Srivatsa. A Handoff Technique to Improve TCP Performance in Next Generation Wireless Networks. – Information Technology Journal, Vol. 7, 2008, No 3.

25. Farooq, M. O., A. B. Dogar, G. A. Shah. MR-LEACH: Multi-Hop Routing with Low Energy Adaptive Clustering Hierarchy. – In: Proc. of 4th International Conference on Sensor Technologies and Applications, (SENSORCOMM'10), 2010, pp. 262-268.
26. Lai, W. K., C. F. Fan, L. Y. Lin. Arranging Cluster Sizes and Transmission Ranges for Wireless Sensor Networks. – Information Sciences, Vol. **183**, 2012, No 1, pp. 117-131.
27. Sha, C., R. C. Wang, H. P. Huang, L. J. Sun. Energy Efficient Clustering Algorithm for Data Aggregation in Wireless Sensor Networks. – Journal of China Universities of Posts and Telecommunications, Vol. **17**, 2010, No 2, pp. 104-122.
28. Fu, S., J. Ma, H. Li, C. Wang. Energy-Balanced Separating Algorithm for Cluster-Based Data Aggregation in Wireless Sensor Networks. – International Journal of Distributed Sensor Networks, Vol. **2013**, 2013. 15 p.