International Journal of Future Generation Communication and Networking
Vol.7, No.2 (2014), pp.21-34
http://dx.doi.org/10.14257/ijfgcn.2014.7.2.03

Fault Tolerant Clustering Protocol for Data Delivery in Wireless Sensor Networks

Deepali Virmani

HOD (IT), BPIT,GGSIPU
deepalivirmani@gmail.com

Abstract

A wireless sensor network (WSN) typically consists of a large number of small sensor nodes with limited energy. Prolonged network lifetime, scalability, node mobility and load balancing are important requirements for many WSN applications. Wireless Sensor networks have limited energy resources so our main aim is to enhance network lifetime by energy balancing. Dynamic clustering plays an important role in enhancing network lifetime. In this paper we propose an idea of fault tolerant optimal path determination for forwarding data to the base station to enhance the network lifetime and implement fault tolerance at the same time. Simulation results prove the betterment of our proposed method over the existing Leach protocol and DCP.

Keywords*: Cluster, Leach Protocol, Network lifetime, Fault tolerance*

I .Introduction

Sensor networks are composed of small, battery operated sensors, whose main function is to collect and forward the required data to the base stations. A sensor network is comprised of sensing, processing, communication ability which enables observation and reaction to events and phenomena in a specified environment [13] [12]. Such a network provides a bridge between the real physical and virtual worlds. Networking of tiny sensor nodes makes it easy to obtain the data about physical phenomena which used to be very much difficult with the conventional ways. Wireless sensor network typically consist of hundreds to thousands of nodes. These nodes collect process and cooperatively pass this collected information to an administrator. The administrator typically is a civil, governmental, commercial, or an industrial entity. The WSN's are a network of nodes that range from hundreds to thousands. Each node is connected to one or several sensors. Sensor nodes are usually resource constrained in terms of energy, processor, memory and bandwidth. The major constraint being limited energy as the sensor nodes are directly dependent on the battery life [1, 11]. Sensor nodes are deployed to gather information and desired that all the nodes works continuously and transmit information as long as possible. This addresses the lifetime problem in wireless sensor networks, so our focus lays on making them live for more amount of time. The base station (administrator) is a master node which is generally fixed and assumed to have unperturbed power supply. The main task of base station is to gather information from various sensor nodes and process it for further dissemination [2].

The sensor nodes are highly resource constrained. Energy is one of the major issues of the sensor nodes. Sensor nodes spend their energy during transmitting the data, receiving and relaying packets. Energy consumption is further increased with the distance, as energy consumption is directly proportional to the square of the distance among the nodes [3]. Hence, designing routing algorithms that maximize the life time until the first battery expires

ISSN: 2233-7857 IJFGCN

is an important consideration. In order to minimize energy consumption and increase lifetime of the network, distance must also be kept under consideration [4, 5]. Scalability being another issue of concern as a sensor consists of hundreds or thousands of nodes [6]. In sensor networks these issues are addressed in cluster based architecture particularly in LEACH [7] and DCP [14]. But Leach supports only static cluster formation where cluster head runs out of energy more quickly due to extra responsibility of data aggregation. Dcp provides dynamic clustering but it does not support optimal path formation and fault tolerance in forwarding data. In this paper we focus on key concern areas with sensor networks like minimizing energy conservation, maximizing network lifetime, real time communication, fault tolerance.

II. Literature Survey

The literature [7] LEACH (Low-Energy Adaptive Clustering Hierarchy) proposed a clustering-based routing protocol that minimizes global energy usage by distributing the load to all the nodes at different points in time. A fixed clustering based approach to prolong the lifetime of sensor networks employing data aggregation is described in literature [4]. The literature [8] RDAG is an approach of Data aggregation by reducing the number of transmissions which is an effective approach to save energy by using the concept of LEACH. The literature [5] and [9] says that energy consumption in a WSN varies with the transmission range. Reducing the transmission range will reduce the power consumed in transmitting a packet toward the neighbours. The literature [14] DCP (Dynamic clustering Protocol) is an approach for dynamic cluster formation method where clusters are refreshed periodically based on residual energy, distance and cost in this sleep and wait technology is also applied to enhance the network lifetime.

The major drawback of LEACH [7] is that it does not support dynamic clustering so to overcome this we use DCP [14] to form clusters. We determine optimal paths for each cluster head to forward its data to the base station. Our proposed method supports fault tolerance in case a conflict occurs or a node goes down and path becomes dead. To best of our knowledge none of the previous work has included all these factors in a single work.

III. Methodology

A. Proposed Fault Tolerant Optimal Path Clustering Protocol (FTOCP)

The proposed FTOCP uses DCP [14] for formation of dynamic clusters. The cluster head behaves as the data aggregating node for each cluster. As soon as nodes forward the data to the cluster head they move to the wait state and remain in the sleep mode until they have something more to transfer [12]. The data when aggregated [8] at cluster head of each cluster is forwarded to the base station The proposed protocol helps in conserving energy by only allowing each cluster head to communicate with other cluster heads to enable formation of an optimal path which will consume the minimum amount of energy to the base station. The optimal path is determined using an assumption that transmission energy consumption will increase greatly as the transmission distance increases. A node which needs to transmit its data sends the data to its neighbour (one of the cluster heads) having the maximum value for a constant determined by using the relation [Energy α (Distance)^2]. The corresponding cluster head forwards the data to its next neighbour having maximum constant value till it reaches the base station. Only the cluster heads with highest energy levels and nearest to base station are used for transmission and the energy of all the other nodes is conserved for future use. The proposed FTOCP also implements fault tolerance as well that is in case one of the nodes in the optimal path fails the sending node immediately looks for the best available path (next node) at that instant which does not include the dead node there by tolerating fault in data transfer. This reduces the frequency of transmission failure.

The salient features of FTOCP include:

- Dynamic clustering
- Reducing number of nodes taking part in transmission
- Fault tolerance
- Minimum energy path for forwarding data
- Reduces communication overhead
- Prolonged Lifetime
- Sleep and wait
- Scalability for large number of nodes
- Energy balancing
- Time to time refresh of clusters

IV. Algorithm

initialise():- Parameters of the node are assigned values here.
cal_distance():- Calculates distance of each node from all other nodes and stores them in the **distance** matrix.

energy_max():- Determines the node with max energy from the nodes which are not a part any cluster yet and returns the **id** of that node.

status_update():- Marks the status of the node as active or inactive.

Make_cluster():

```
Loop1: For i=1 to No_of_Nodes
    Do
        Emax_id=energy_max()
        if(emax_id!=-1) then
            subsink[i]=node[emax_id].id
            node[emax_id].ss_flag=1
            increment cluster_id
            node[emax_id].cluster_no= cluster_id
            loop2: For j=1 to No_of_Nodes
                Do
                    if(node[j].cluster_no==0) then
                     if(distance[emax_id][j]<=range) then
                          node[j].cluster_no=cluster_id
            end loop2
end loop1
```

In this algorithm the node with the highest energy level looks for nodes within its transmission range to form a cluster and appoint itself as the cluster head of the cluster formed. The process goes on till all the nodes are not covered under some or the other cluster.

int minimum(int a[],int m[],int n)

```
 min=9999;
t=0;
Loop: For i=1 to n
   Do
  if(m[i]!=1) then
     if(mi>=a[i]) then
min=a[i];
t=i;
 end Loop
return t;
```

In this algorithm, the minimum cost is determined and node with minimum cost is returned.

void printpath(ints,inti,int p[])

```
if(i==s)
print BS_node;
trace[head][++pathnode]=BS_node;
else if(p[i]==0)
 print “no path from BS_node to cluster head[i-1]”;
else
printpath(s,p[i],p);
print cluster head[i-1];
trace[head][++pathnode]=cluster head[i-1];
```

In this algorithm, the path from base station to the cluster head is printed and also stored inside the trace matrix.

path_maker()

```
     head=0,pathnode=0,counter=0, num_of_vertices= no_of_clusters +1;
     Loop1: For i=0 to (num_of_vertices -1)
          Loop2: For i=0 to (num_of_vertices -1)
            Do
    distance_clusterheads[i][j]=distance between cluster head[i] and
                              cluster head[j];
    end Loop2
     end Loop1
     Loop: For i=0 to (num_of_vertices -1)
           Do
          distance_BS[i]=distance of cluster head[i] to base station
     end Loop
     Loop1: For i=0 to (num_of_vertices -1)
          Loop2: For i=0 to (num_of_vertices -1)
            Do
                  if(distance_cluster heads[i][j]<=transmission_range) then
          adjacency[i][j]=1;
          graph[i+1][j+1]= distance_BS[j];
          else
          adjacency [i][j]=0;
    graph[i+1][j+1]=9999;
    end Loop2
     end Loop1
     source= no_of_clusters +1;
     Initialize matrices mark[ ]=0, pathestimate[ ]=9999, predecessor[ ]=0;
    pathestimate[source]=0;
     while(counter<num_of_vertices)
     Do
    u=minimum(pathestimate,mark,num_of_vertices);
    mark[u]=1;
          Loop: For i=1 to num_of_vertices
            Do
                   if(graph[u][i]>0) then
          if(mark[i]!=1) then
    if(pathestimate[i]>pathestimate[u]+graph[u][i]) then
          pathestimate[i]=pathestimate[u]+graph[u][i];
          predecessor[i]=u;
          end Loop
          end while
  Loop: For i=1 to num_of_vertices
   Do
  ++head, pathnode=0;
  printpath(source,i,predecessor);
```

```
if(pathestimate[i]!=9999)
   Print  pathestimate[i];
   end Loop
```

In this algorithm, the path from cluster heads to the base station is formed. The cluster head send its data to another cluster head which is within its transmission range and as well as nearest to the base station. This process is repeated till the path to base station is completed.

FTOCP():

```
while true,
   do
while(cluster head[n]!=0)
   do
            if(energy_val[cluster head[n]-1]<=0) then
               print NETWORK DEAD
              return false
            decrement energy_val[cluster head[n]-1] by 10 units
            increment n
end while
   loop1: for time=1 to REFRESH_TIME
   do
             wait for 1 time unit
             update_status();
             loop2: for i=1 to No_of_Nodes
                do
                if(node[i].energy<=0) then
   print NETWORK DEAD
   return false
               if(node[i].ss_flag!=1) then
               if(node[i].active==1) then
               decrement energy_val[i] by 2 units
                           active_status[i]=0;
                           node[i].active=0;
                else
                           decrement energy_val[i] by 1 unit
              end loop2
               end loop1
      cluster_id=0;
               initialise();
      cal_distance();
      make_cluster();
      path_maker();
   redisplay output
   end while
```


In this algorithm transmission of data takes place from all the other nodes to the cluster head. Cluster head behaves as the data aggregating node for that particular time interval. As soon as nodes forward the data to the cluster head they move to the wait state and remain in the sleep mode until they have something more to transfer. The cluster heads then transfer the data to the base station through an optimum path. After some interval of time called refresh time clusters are reformed and the process repeats.

V. Working Scenario

A. Optimal Path Formation

Cluster heads: filled circles ●
Nodes: empty circle ○
Optimal Path: dark line ▬
The different colours represent the different clusters formed.

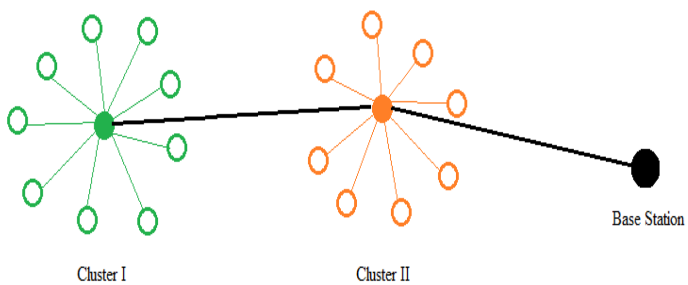

Figure 1. Cluster Representation

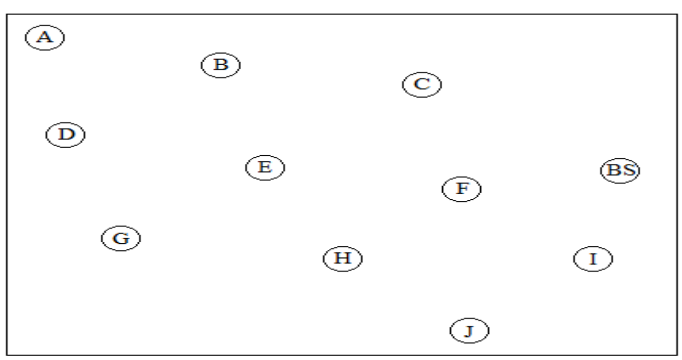

Figure 2.1. Initial Configuration

Figure 2.1 Shows initial configuration that is 10 cluster heads namely A, B. . . J. There is a base station BS. We need an optimum path between every cluster head and base station.

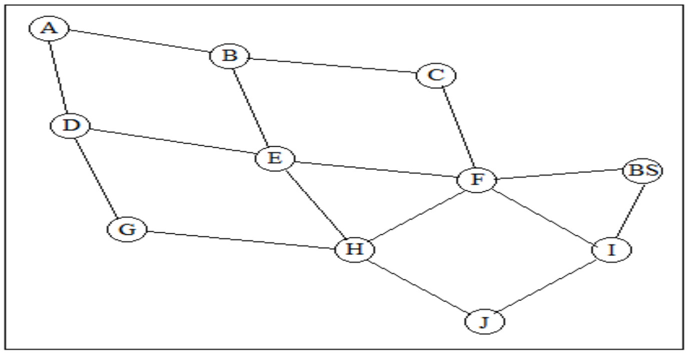

Figure 2.2. All Possible Links between Nodes

Figure 2.2 shows all possible links between the nodes. Each node has a fixed transmission range up to which it can send the data. So, there can a link between two nodes if are in each other's transmission range.

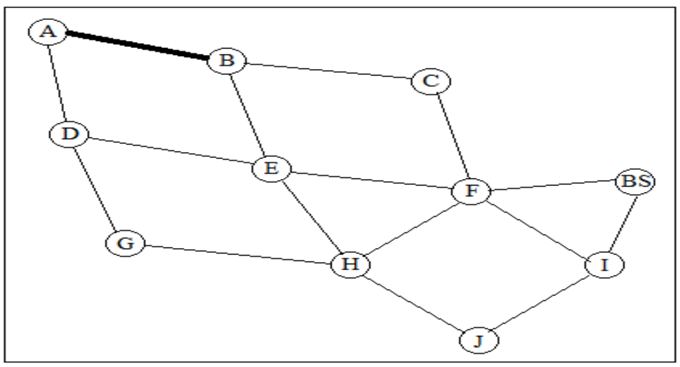

Figure 2.3. Optimal Neighbour Selection

Figure 2.3 shows that in finding path from A to BS, A has links to two nodes B and D. It will select that node which consumes minimum energy. Since B consumes less energy than D, it selects B.

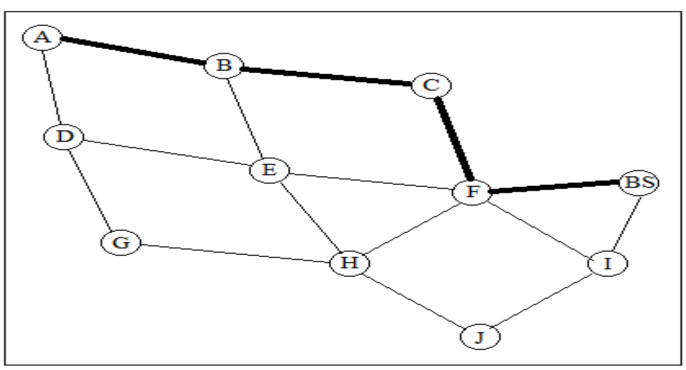

Figure 2.4. Optimal Path Formation

Figure 2.4 shows that in finding path from A to BS, A selects B. Similarly from B, node consuming minimum energy is selected *i.e.*, node C. The process is repeated until we reach the final node i.e. BS and optimal path is formed.

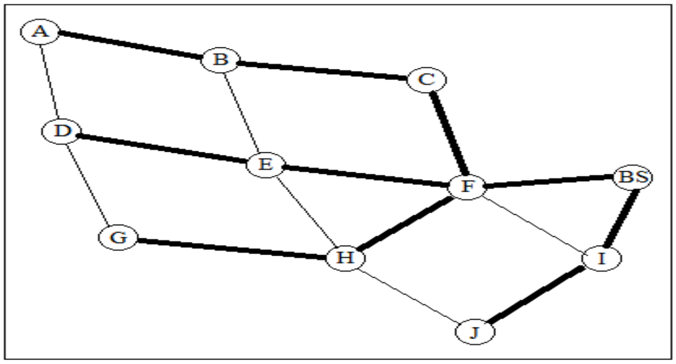

Figure 2.5. Final Configuration

Figure 2.5 Shows final configuration that is 10 cluster heads namely A, B. . . J. Each connected to the BS through an optimal path that consumes minimum energy.

B. Fault tolerance

Dead Path: ——x—

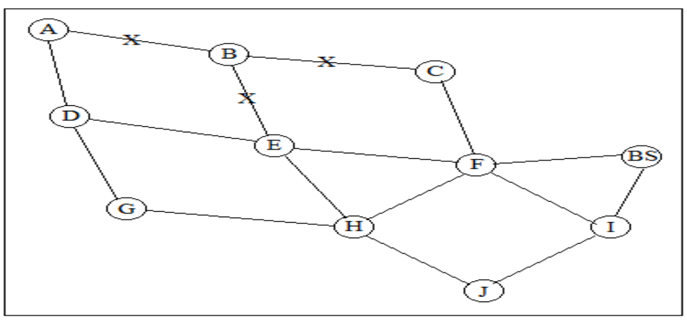

Figure 3.1. Nodes Going Down

Figure 3.1. shows node B, E and C going down and hence the paths A-B, B-C, B-E become dead.

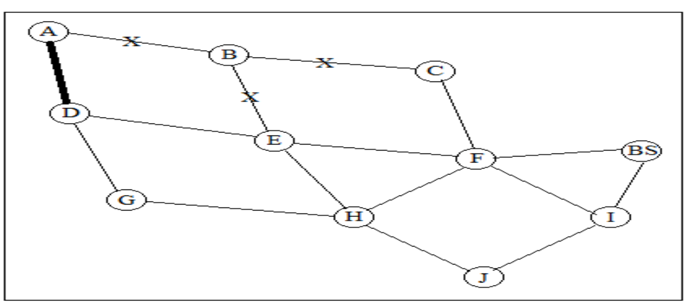

Figure 3.2. Secondary Neighbor Selection

Figure 3.2 shows that in finding path from A to BS, since the optimal path (A-B) is dead. It looks for the secondary neighbour from A *i.e.*, node D and selects it.

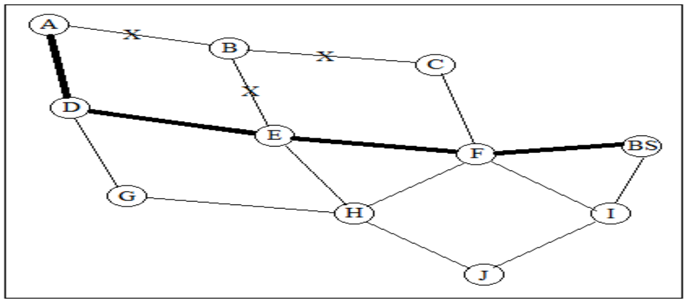

Figure 3.3. Fault Tolerance

Figure 3.3 shows implementation of fault tolerance in finding path from A to BS, since the optimal path (A-B) is dead. It looks for the secondary neighbour from A *i.e.*, node D and selects it. Now from D, it will look for the optimum path to reach the base station.

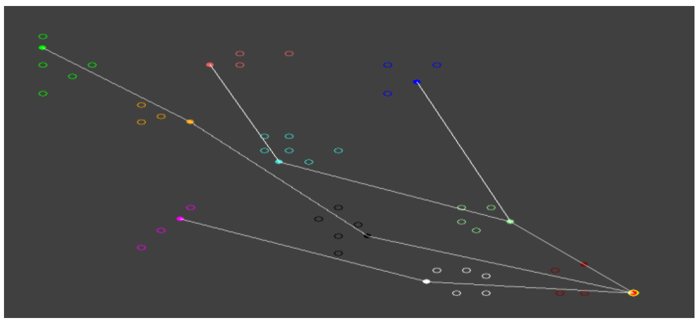

Figure 4.1. Optimal Path Formation along with Clustering

Figure 4.1 represents the formation of clusters along with best path available for each cluster head to forward its data to the base station via other cluster heads. This path always consumes minimum amount of energy for forwarding data.

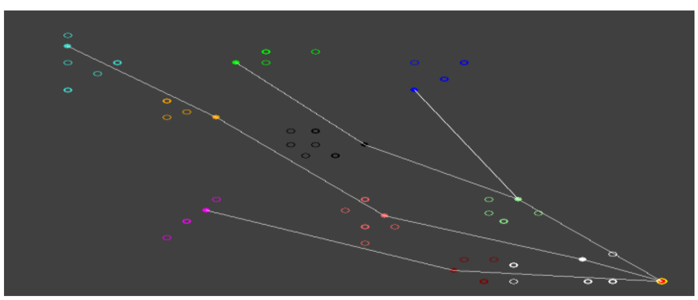

Figure 4.2. New Clusters and Path after Refresh Time

Figure 4.2 represents the nodes and path to the base station for each cluster head after refreshing time when new clusters are formed and new cluster heads are appointed.

VI. Simulation Results

We simulated and evaluated the performance of the proposed protocol with the existing Leach. All the simulations are done in MATLAB in order to validate the efficiency of proposed protocol.

Table 1. Simulation Parameters

No.	Parameter	Value
1.	Simulation Area	1000x1000
2.	No. of nodes	450
3.	Radio Propagation Model	Two way ground
4.	Channel Type	Wireless Channel
5.	Antenna Model	Antenna/Omni antenna
6.	Energy Model	Battery
7.	Round Duration Time	10s
8.	Initial Energy of each node	0.5J

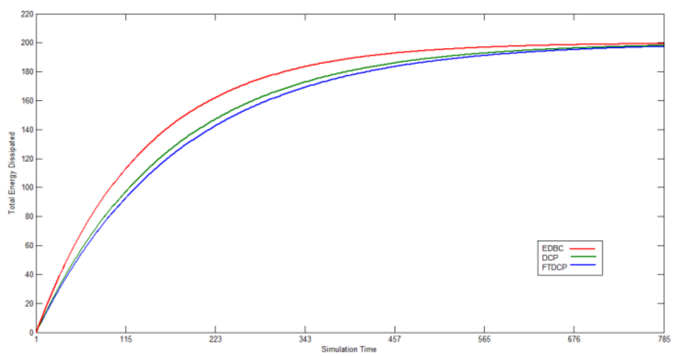

Figure 5.1. Total Energy Dissipated in EDBC, DCP and FTOCP

Figure 5.1 shows the total energy consumption of the network over simulation time. The simulations prove FTOCP to be better than EDBC and DCP in terms of energy consumption. At the start of simulation, performance of all are comparable but as the simulation time proceeds FTOCP conserves a lot of energy due to its sleep wait and fault tolerant technology. As only the active nodes consume energy and nodes after forwarding data to cluster head go to sleep mode.

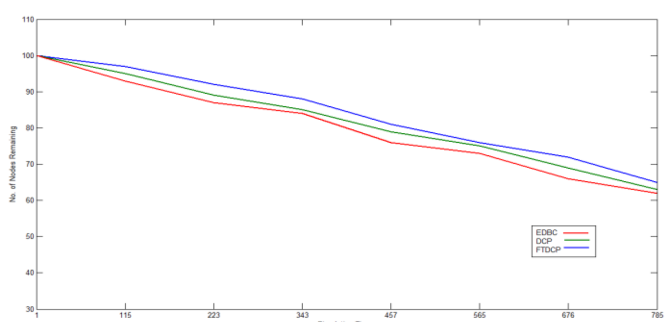

Figure 5.2. Node Lifetime

Figure 5.2 shows the total number of nodes running in the network over simulation time. The simulations prove FTOCP has lesser number of dead nodes than EDBC and DCP. At the start of simulation number of nodes running is same for all but as the simulation time proceeds node failure is lesser in FTOCP sleep wait and fault tolerant technology.

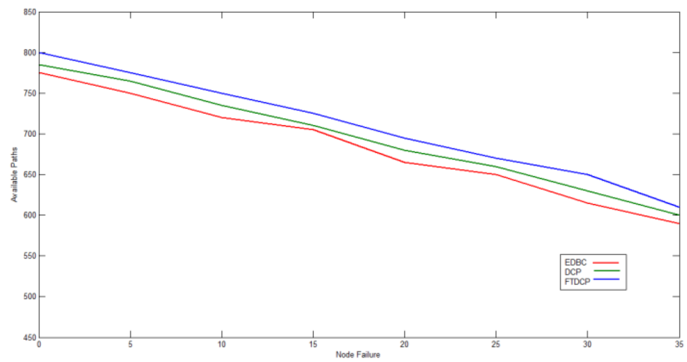

Figure 5.3. Evaluation of Fault Tolerance

Figure 5.3 shows the available paths of nodes running in the network over simulation time. The simulations prove FTOCP has lesser number of failures than EDBC and DCP. At the start of simulation available paths for all are comparable but as the simulation time proceeds possible available paths decrease slowly as compared to EDBC and DCP due to fault tolerance in FTOCP.

Table 2. Comparison among Protocols

Evaluation Parameter	LEACH	RDAG	EDBC	DCP	FTOCP
Clustering	Static	Static	Static	Dynamic	Dynamic
Energy Efficiency	Very Low	Low	Moderate	High	Highest
Delay	Least	High	Moderate	High	Highest
Network Life	Low	Low	High	High	Highest
Fault Tolerance	No	No	No	No	Yes
Packet Loss	Maximum	Moderate	Less	High	Minimum

Table 2 shows that performance of our proposed protocols DCP and FTOCP is better than existing protocols in mostly all aspects like energy efficiency, network lifetime, etc. However, the delay parameter of proposed protocols is on the higher side but its value is kept within the real time delay parameter so that it does not affect the overall performance.

VII. Conclusions

Clustering is a useful topology-management approach to reduce the energy consumption and exploit data aggregation in wireless sensor networks. In this paper we have focused on dynamic clustering where clusters are refreshed periodically and cluster head is selected accordingly. The existing protocols LEACH and RDAG support static clustering and cluster head is fixed in the entire scenario. Cluster head being used as the data aggregator node runs out of energy. So in this paper we have proposed a dynamic clustering protocol (DCP) that supports dynamic clustering with support of sleep and wait technology, where the node that needs to transmit the message is only in wake state after forwarding the message the node changes the state to sleep. By this a lot of energy is conserved enhancing network lifetime indirectly. We have simulated and compared the results in MAT lab of our proposed protocol DCP with the existing LEACH. Simulation results our proposed protocol is better in terms of energy conservation and enhancement of network life time as sleep and wait technology enhances the lifetime of nodes. Delay is increased in our DCP but still it is in real time communication limits so increased delay is not a harmful factor.

References

[1] M. Saeidmanesh, M. Hajimohammadi and A.Movaghar, “Energy and Distance Based Clustering: An Energy Efficient Clustering Method for Wireless Sensor Networks”, World Academy of Science, Engineering and Technology, vol. 31, **(2009)**.
[2] J. N. Al-Karaki, R. Ul-Mustafa and A. E. Kamal, “Data aggregation in wireless sensor networks- exact and approximate algorithms”, Proceedings the International Workshop on High-Performance Switching and Routing, Phoenix, AZ, **(2004)** April.
[3] N. Rajani, “Energy Efficient, Reliable Routing Protocol for Wireless Sensor”, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, **(2008)** June.
[4] I. F. Akyildiz, W. Su, Y. Sankara Subramaniam and E. Cayirci, “Wireless sensor networks: a survey”, Published by Elsevier Science B.V, Computer Networks, vol. 38, **(2002)**, pp. 393-422.
[5] W. RabinerHeinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Micro sensor Networks”, Proceedings of the 33rd Hawaii International Conference on System Sciences, **(2000)**.
[6] G. Mohammad, H. Yeganeh, N. Alinaghipour and A. Movaghar, “RDAG: A Structure-free Real-time Data Aggregation Protocol for Wireless Sensor Networks”, 17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.
[7] J. Zhang, Q. Wu, F. Ren, T. He and C. Lin., “Effective data aggregation supported by dynamic routing in wireless sensor networks”, Proceedings of the IEEE International Conference on Communications (ICC’10), **(2010)** May, pp. 1-6.
[8] C. Jaikaeo and C. Chien-Chung Shen, “Sensor Information Networking”, **(2000)**.
[9] Y. Kim and X. Grivo, “Performance Analysis on Energy Aware MAC protocol for Wireless Sensor Network”, 2011 International Conference on Network and Electronics Engineering IPCSIT, © (2011) IACSIT Press, Singapore, vol. 11, **(2011)**.
[10] A. Xavier Das, C. Eapen, A. Ashok, S. Tripathi and R. MabrukShadi, “Dynamic Event Based Energy Efficient Routing Protocol For Wireless Sensor Networks (WSNs)”, International Journal of Engineering and Advanced Technology (IJEAT)ISSN: 2249 – 8958, vol. 1, no. 6, **(2012)** August.
[11] A. Bharathidasan, V. Anand and S. Ponduru, “Sensor Networks: An Overview”, Department of Computer Science, University of California, and Davis, CA 95616.
[12] A. A. Abbasi and M. Younis, “A Survey on Clustering Algorithms for wireless sensor network”, Computer Communications, vol. 30, **(2007)** June 21, pp. 2826-2841.
[13] K. Sohraby, D. Minoli and T. Znati, “Wireless Sensor Net-Works Technology, Protocols, and Applications”, John Wiley, New York, **(2007)**.

[14] D. Virmani, A. Jain, A. Khandelwal, D. Gupta and N. Garg, "Dynamic Clustering Protocol for Data Forwarding in Wireless Sensor Networks", International Journal of Computers & Technology, vol. 7, no. 2, **(2013)**, pp 549–557.

