Energy Efficient Data Retrieval in Wireless Sensor Networks for Disaster Monitoring Applications

Kavitha Kayiram
Dept. of CSE,
Gokaraju Rangaraju Institute of Engineering and Technology,
Hyderabad, India
kavitha.bits@gmail.com

Dr. P.Chandra Sekhar Reddy
Dept. of CSE,
Gokaraju Rangaraju Institute of Engineering and Technology,
Hyderabad,India
pchandureddy@yahoo.com

Dr. Avinash Sharma
Dept. of CSE,
M.M. Engineering College,
Maharishi Markandeshwar (Deemed to be University)
Mullana, Ambala, India.
asharma@mmumullana.org

Dr. R.V.S.Lalitha
Dept. of CSE,
Aditya College of Engineering and Technology,
Kakinada, India.
rvslalitha@gmail.com

***Abstract*—Occurrence of natural disasters is a major concern for people residing in disaster prone areas and this increases the need for Disaster Monitoring. Wireless Sensor Network (WSN) technology offers a better solution in disaster monitoring to facilitate effective rescue operations. WSNs have the capability of quick sensing and transmission of critical data and thus prove effective in reliable monitoring and rescue operations. As the networks are deployed in disaster prone regions, the power, storage resources are limited and data retrieval is the major concern. Many schemes like sleep-scheduling, collaborative sensing have been implemented to solve these issues. But, there are many limitations to enhance the network lifespan, network traffic and effective data retrieval associated with it. In this paper, we propose a novel collaborative sensing scheme along with duty cycling to enhance the lifetime of the network. Our mathematical evaluation prove the energy efficiency of our scheme for disaster applications in WSN.**

Keywords—Wireless Sensor Network, Collaborative Sensing, Disaster Monitoring System, Sleep-scheduling, Duty Cycling.

I. INTRODUCTION

Disasters occur in the nature for various reasons but the consequence of a disaster disturb natural vegetation, human life causing heavy loss. Hence, it is desirable to detect the disaster and warn early to plan timely evacuation of the space and save human life. However some disasters give a late warning with high intensity such as Tsunami, floods etc. and there is very less time for precautions and safety measures. These early warning systems generate alarm signals to indicate the dangers associated with disasters. It is highly desirable for rescue system to get the precise location to launch timely rescue operations. Hence, timely reporting and data backup during the event is important for reducing the victims and damages in incidents. Another major issue is that the disaster may sometimes lead to damage in the network both in terms of hardware and communication medium. This makes it difficult to obtain information about the event and to plan an effective data backup from the network. The Wireless Sensor Network (WSN) [1] offers a better solution to the afore-mentioned scenarios with their cutting edge technology. Typical applications of WSN in disaster monitoring systems include Mine Fire warning system, Earth-Quake Detection system [2], Land-slide Detection system, Volcano monitoring system, Fire detection system etc.

The WSN technology can be used in disaster monitoring to facilitate early warning mechanism. In general the sensor nodes of a WSN are deployed in the region prone to disaster for continuous monitoring and event detection. The sensor nodes monitor the physical phenomenon and generate an early warning in case it senses any unusual behavior in the environment. The sensor nodes are meant to detect certain parameters in the environment like humidity, pressure, temperature, sound intensity, etc. Depending on the application, these sensor nodes are deployed in the disaster prone region to monitor the environment continuously. Each sensor node is equipped with a battery as power source and it consumes this energy to perform sensing, transmission and reception operations in a continuous manner. The lifetime of the battery is limited. As the networks are deployed in remote geographic locations and it is impossible to recharge or replace battery if battery is completely exhausted. We need to conserve the battery power by minimizing the data transmission activity in the network. Hence we propose to use duty cycling in WSN to minimize the data transmissions in the network while also catering to the goals of disaster monitoring.

Many energy efficient schemes were proposed in WSN for optimal power utilization like data aggregation [3], data compression [4], and load balancing [5], sleep-scheduling etc. Each sensor node has a built-in storage unit, processor, sensing unit, battery, transmitter and receiver. The sensor node can sense data and store in the storage unit and whenever there is an external query from the Base-Station (BS)

978-1-7281-5681-1/21/$31.00

the query is processed and the result is sent to the BS. Although data compression algorithms are used to compress the sensed data at the node itself. For applications of WSN, multiple sensor nodes are involved in Query processing, the query results would have been from multiple sensor nodes, so aggregated data is sent to the BS. The data aggregation means to perform the aggregation function (sum, min, max, average etc.) on the data so as to reduce the data size which will eventually reduce the traffic in the network and as well as the communication load on the sensor node. In general all the WSN applications, require redundant deployment of sensor nodes, which leads to redundant data in the network. The sleep-scheduling mechanism will conserve battery power by turning off the radio when not in use. This will aid to energy efficiency in the network.

The initial deployment of sensor nodes is important in achieving energy efficiency in WSN. One of the primary design aspects in WSN is connectivity in the network. Hence, the deployment of nodes is densely done to maintain connectivity. Many WSN applications require a dense deployment of nodes. Due to this deployment strategy, the sensor nodes exhibit redundancy in their collected data. (Also due to spatio-temporal correlation [6] found in sensor data, there is certain amount of redundancy) If all the nodes are active at the same time, it will dissipate energy fast and shorten the network lifetime. Therefore, it is essential to make some of the nodes sleep, while others are active and sensing. Hence, a significant amount of energy savings can be achieved by turning off some nodes, which will eliminate redundant data. There has been various sleep-scheduling schemes proposed to suit different scenarios. By maintaining sufficient coverage and connectivity, the sensor nodes employ suitable duty-cycling schemes to conserve the network power.

In this paper, we propose an energy efficient collaborative sensing scheme for effective disaster monitoring using WSN with mesh topology. Our approach is based on the following concepts.

1. A novel collaborative sensing scheme by creating a cluster of nodes,
2. To reduce the amount of energy exhausted by individual node using sleep-scheduling in the cluster.
3. The data retrieval in case of disaster or partial damage in the network.

We outline the latest work done so-far in the area of Collaborative Sensing in WSN, in Section II. Our proposed scheme is described in Section III. Then we evaluate the performance of the proposed approach in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

In this Section, we give a brief description of the research work done in the collaborative sensing and duty cycling in WSN along with the schemes adopted for disaster monitoring.

The initial deployment of sensor nodes is important in achieving energy efficiency in WSN. One of the primary design aspects in WSN is connectivity in the network. Hence, the deployment of nodes is densely done to maintain connectivity. Many WSN applications require a dense deployment of nodes. Due to this deployment strategy, the sensor nodes exhibit redundancy in their collected data. (Also due to spatio-temporal correlation found in sensor data, there is certain amount of redundancy) If all the nodes are active at the same time, it will dissipate energy fast and shorten the network lifetime. Therefore, it is essential to make some of the nodes sleep, while others are active and sensing. Hence, a significant amount of energy savings can be achieved by turning off some nodes, which will eliminate redundant data. There has been various sleep-scheduling schemes proposed to suit different scenarios. By maintaining sufficient coverage and connectivity, the sensor nodes employ suitable duty-cycling schemes to conserve the network power.

The work in [7] has proposed a method to estimate the number of nodes to be deployed to ensure a predetermined lifetime. First the deployment area is divided into equal sized strips and nodes are deployed in such a way that the density of deployed sensors increases as the distance between a strip and the sink decreases. In a tree topology, it is a well-known fact that the nodes nearer to the sink node (intermediate nodes) consume more energy than the leaf nodes. Hence, there is a requirement of dense deployment in the strip nearer to the sink than the strip farther from the sink. The energy consumption in the network is computed using the energy dissipation model [8]. But, this method neither considers the coverage issues nor handles the node failure situations. Further this work does not consider the energy dissipation pattern for data forwarding sensor networks with sleep based energy saving support.

Collaborative sensing of sensor nodes has been a very significant technique to utilize the power at a node and transmit critical data for a prolonged time period in an optimal way. Various Collaborative schemes have been proposed with both tree topology and graph topology. As the tree topology schemes have a critical limitation that the child nodes (usually the storage nodes) have only a single path to transmit their data. During a critical disaster event there is negligible possibility of data retrieval if the intermediate nodes fail. Hence, we concentrate on the recent works of mesh topology.

A gradient based sensor deployment scheme has been proposed in [9] to assure the complete coverage and maximum network lifetime. The major drawback of the work is that it assumes regular terrain and does not consider the pattern of data forwarding tree. It can be noted that the existing schemes do not consider all aspects of deployment design like potential irregularity in terrain, gradient in traffic load distribution for tree-based [10] data gathering, requirement of redundant sensors.

There had been various schemes for event detection in disaster monitoring applications of WSN. We focus on event detection done at node level with a pre-fixed threshold value specific for applications of WSN. The work in [11] proposed an in-network decentralized approach for event detection in WSNs based on machine learning techniques. This approach adopted decision trees for distributed event detection and a reputation-based voting method for detected results aggregation over the sensor nodes in order to reach a consensus decision. The work makes use of decision trees and achieves highly accurate results in terms of detection accuracy. Although the time complexity of the machine learning techniques used to design this approach was thoroughly discussed, the communication overhead which is the main player of energy consumption was not investigated Sleep-scheduling of sensor nodes has been a very significant technique to utilize the power at a node in an optimal way. As mentioned earlier according to this approach, a node will go to sleep mode for some time thus consuming minimal power.

The work in [12] proposes a protocol for rare-event detection and not for continuously reporting readings from the sensor network. As there is no continuous data transmission the nodes need not adopt a synchronized sleep-scheduling, which will lead to energy wastage. Initially, each node randomly selects a wakeup time and communicates it to its neighbors. In the following iterations, each node recalculates its wakeup time exactly once per iteration, based on the most recently updated neighbor schedules. Therefore, at any point of time the network has a partial sensing coverage. This is done by turning on a subset of nodes to form primary nodes sensing and covering the network for some time and then wakeup their neighbors and go to sleep to improve network lifetime. Further each sensor node is duty cycled in co-ordination with its neighbors such that the average detection delay is minimized. For a rare event detection the sensor nearby detects the event and transmits to the sink via the intermediate nodes. As the network is an unconnected graph at times, due to the duty cycling of nodes non-uniformly, there is a time delay in the transmission of this event.

In one of the research works, a data storage scheme called Distributed Index-Based Dominating Set (DIDS) [13] is proposed to efficiently process data queries. This scheme proposes, that sensed data is stored at the node close to the event detecting nodes and the location information of these storage nodes is pushed to some index nodes close to them. So the queries are only routed to the index nodes instead of flooding the whole network. Therefore DIDS uses limited network and computational resources while providing timely responses to queries.

Now, we list out the limitations of the existing schemes mentioned in the preceding discussion. This scheme, is based on hierarchal distribution on nodes. Though it strictly does not follow the tree topology but after the cluster formation is done, several parts of the network shall have a tree topology obtained and hence the limitation of data retrieval during a disaster surfaces back. Also as the intermediate nodes are the ones which lose the most amount of energy when data is transmitted back, in this scheme the intermediate nodes are the storage nodes and hence we can expect a greater loss of data.

Now, we present our proposal in Section III, followed by the mathematical evaluation of our scheme to prove its efficiency in Section IV.

III. THE PROPOSAL

The proposed collaborative sensing scheme is based on the mesh topology and in this work we propose three modules:

1. Deployment
   - Cluster Formation
   - Twins Formation
2. Sleep-scheduling
3. Data retrieval on node failure

A. Deployment

In this model, the sensor nodes are deployed in the long tunnels of the mine. To address node failure in WSN caused due to fire disasters, we place multiple sensor nodes in the same region. This may lead to sensor nodes sensing the same physical phenomenon thus causing redundancy. Redundancy ensures backup data in case of node failure. We also employ sleep-scheduling model to achieve energy efficiency in WSN.

(i) Cluster Formation

All the nodes which sense similar temperature readings form a cluster. The Sensor nodes are deployed in the area of interest. Each node will sense data for a certain time period. One of the nodes will initiate the clustering activity and is termed as the coordinator node. The coordinator will send a cluster request along with its sensed data to all its one hop neighbors. Each node compares this '*index compare'* data with their own readings and if the node sensed data is similar, then the node sends an '*ack*' back confirming to be part of the cluster as shown in Figure 1. If the coordinator node receives '*ack*' from a unique node id, then the cluster counter (k) is incremented by one. This new member of the cluster starts sending the cluster request to its one hop neighbors. The process continues till the cluster counter is either 8 or when the first dis-similarity in data is observed. If the coordinator node counter reached 8 or receives rejection (whichever is first) then the node which has sent the acknowledgement/rejection becomes the coordinator for the next cluster and the procedure is repeated till all the nodes in the network are covered as shown in Figure 2.

The number of members in a cluster range from 1 to 8. Depending on the tunnel dimensions, the number of sensor nodes with in the proximity of a coordinator vary. Also, if the sensors exhibit dis-similar data then they cannot be part of the cluster. Hence, a dis-similar reading will lead to a different cluster formation. The maximum size of the cluster is fixed as 8 because with an increase in this size will lead to high redundancy and communication cost constraints. For any cluster size less than 8, the nodes work for longer duration leading to decrease in lifetime. The detailed algorithm is given as follows:

Algorithm:

1: Choose a root, r Ɛ V (using global Leader Selection Algorithm)
2: Root r becomes the first coordinator node c.
3: c has a counter k=1, k1=0; every node has an index compare counter i=0;
k = increments when a node sends acknowledgement.
k1= increments when a node sends nack.
i = 1; when the index compare data of c is similar to that of the node
i = -1; when the index compare data of c and the node is dissimilar
i = 0; when the index data of c is not yet compared to that of the node.
3: Compute the hop distance from r to each node (using Breadth First Search (BFS))
4: n = c;

```
while (k<=8 && k1<1)
{
    if (n has neighbor nodes)
    {
      Transmit index compare data to all 1 hop neighbors if i=0;
      All nodes with i =1, return ack to c (for every ack k++);
      All nodes with i =-1, return nack to c (for every nack k1++);
     Every individual node with i = 1 becomes n and follows this iteration.
     All i=1 nodes form a cluster;
    }
}
```

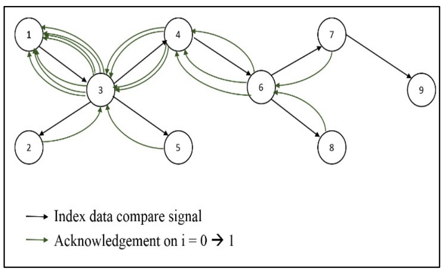

Fig. 1 Cluster formation, with k=8

Assumption:
1: Nodes 1, 2, 3, 4, 5, 6, 7, 8 have similar data readings
2: Only diagonals are 1 hop distance.
3: 1→r (root node and 1st coordinator node)

States:

1: 1→r	{k=1, k1=0 ;}
2: 1→3	{i = 0→1; 3 Ɛ n; 3 sends ack; k++ (k=2)}
3: 3→2	{i = 0→1; 2 Ɛ n; 2 sends ack; k++ (k=3)}
3→4	{i = 0→1; 4 Ɛ n; 4 sends ack; k++ (k=4)}
3→5	{i = 0→1; 5 Ɛ n; 5 sends ack; k++ (k=5)}
4: 4→6	{i = 0→1; 6 Ɛ n; 6 sends ack; k++ (k=6)}
5→6	{i = 1;}
5: 6→7	{i = 0→1; 7 Ɛ n; 7 sends ack; k++ (k=7)}
6→8	{i = 0→1; 8 Ɛ n; 8 sends ack; k++ (k=8)}
6: 7→9	{k>8; 9→new coordinator node}

Assumption:
1: Nodes 1 to 6 have similar data readings and 7, 8 has dissimilar data.
2: Only diagonals are 1 hop distance.
3: 1□r (root node and 1st coordinator node)

States:

1: 1□r	{k=1, k1=0;}
2: 1□3	{i = 0□1; 3 Ɛ n; 3 sends ack; k++ (k=2)}
3: 3□2	{i = 0□1; 2 Ɛ n; 2 sends ack; k++ (k=3)}
3□4	{i = 0□1; 4 Ɛ n; 4 sends ack; k++ (k=4)}
3□5	{i = 0□1; 5 Ɛ n; 5 sends ack; k++ (k=5)}
4: 4□6	{i = 0□1; 6 Ɛ n; 6 sends ack; k++ (k=6)}
5□6	{i = 1;}
5: 6□7	{i = 0□-1; 7 sends nack; 7 is new coordinator node}
6□8	{i = 0□-1;}

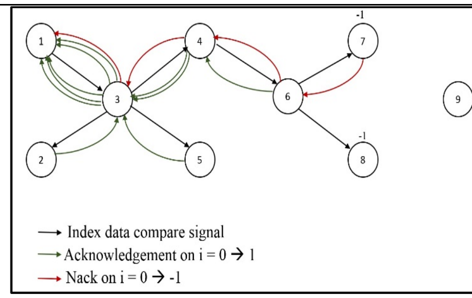

Fig. 2 Cluster Formation, with i = -1

The first node with i=-1 or when k>8 (whichever occurs first) becomes the coordinator node for the new cluster and process continues till all the nodes of V are clustered.

(ii) Twin Node selection within a Cluster

The nodes in the cluster operate in twin-mode with one node being a *'primary node'* and the other node *'secondary node'*. The role of primary node is to sense and transmit the data to the BS and secondary node senses and stores the data for back up. In this way, we have fewer data transmissions and redundancy elimination to BS. The node closest to the BS is termed as link node. The link node by default is a 'primary node'. For a cluster of size 'n', it can have 'm' number of primary nodes where m is computed as follows:

m= n/2; if n is even.
m=Integer (n/2) +1; if n is odd.

The m-1 nodes closest to the link node become the primary nodes. The primary nodes looks for its twin (nodes other than primary) which are close to them. Priority of twin formation depends on the distance of the primary node from the link node as shown in Figure 3. The farther the primary node the less the priority to be a twin node. If the number of nodes in a cluster are not even then the primary node with the lowest priority does not have a secondary node and hence acts as an independent node in the cluster.

In case of partial damage in the network, leading to failure of some of the nodes in the cluster, we will be able to retrieve the critical data from its twin node. After the failure of primary node, the secondary node becomes the independent primary node and will take up the responsibility of transmitting data to BS.

Algorithm:
1: Choose a link node (the closest node in the cluster to r, using BFS)
2: n = no of nodes in the cluster;
m = no of primary nodes;
n1= no of secondary nodes;
i=1, j=1;
3: Every node has a primary counter p=0;
And a secondary counter s=0;

```
4:   if (n == even)
         m=n/2;
     else
         m=n/2+1;
```

5: The m-1 nodes closest to the link node become primary nodes; higher the priority number starting from 2.

```
t=link node;
while (m>0)
{
```

```
  Transmit to all the 1 hop distant cluster member;
   If (p==0)
   {
   t= 1 hop neighbor
   P=i++;
   m--;
   }
}
6: Link node becomes the first primary node; (priority=1)
7: The remaining nodes are termed secondary nodes. The higher priority
number starting from 1.
t= the node which got m value as 0; s=1;
while (n1>0)
{
  Transmit to all the 1 hop distant cluster member;
   If (s==0)
   {
   t= 1 hop neighbor
   s=j++;
   n1--;
   }
}
8: The primary and secondary nodes with same priority number are
forming twins to follow same sleep schedule.
9: The node p and s are primary and secondary priority of each node.
```

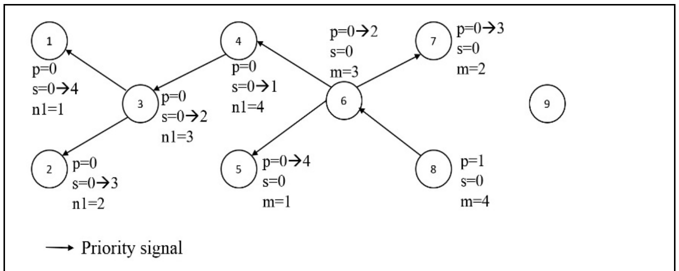

Fig. 3 Twin node selection, for k=8

B. Sleep-scheduling of Nodes

Sleep-scheduling basically defines the time slot for which a twin/independent node is awake and senses the data. Once twins are formed, two nodes in the twins will have same sleep-schedule. Data is transmitted by the primary nodes of the cluster three times in every 24 hours (once every 8 hours). Hence the 8 hours is divided into 'm' slots and each slot is assigned to the primary nodes (both the twin and the independent node) according to their priority (higher the priority lower the slot number). In the assigned time slot the twins/independent node sense and store data every 1 min and the data transmitted at the end of 8 hours to the base station. The data sent is the aggregate computed on the stored data. Thus the twins/independent primary nodes sense their data in their time slots in periodic cycles.

The link node or the first primary node aggregates the data from the other primary nodes and transmits it during its activity time. The same is followed by the other primary nodes in the cluster during their activity time.

C. Data Retrieval on Node Failure

(i) Critical Event Detection

If the difference between two consecutive readings of a twin/independent node exceeds the defined threshold value then the system goes to critical event detection state. In this state the twin/independent node activates the other twin/independent nodes and they immediately start transmitting the data to the base station irrespective of their sleep-scheduling slots. The twin/independent node also sends an alert signal to the BS.

(ii) Rescheduling

If the primary node fails due to any other reason other than critical event detection, then sleep-scheduling of nodes depends on whether the primary node was one of the twin or independent node.

Case 1: If the primary node is one of the twin node, in this case, if the secondary node is still working then the sleep-scheduling of the cluster is intact and the secondary node now becomes the independent primary node. If the secondary node of the twins fails then the immediate next primary node in priority is triggered to immediately start sensing (even though it is not its slot to sense) and the sleep schedule is rescheduled. When rescheduling occurs, the time slot of each twin/independent node is now redefined. Each twin/independent node senses data for a time slot of 8/(m-1).

Case 2: If the primary node is independent, in this case, there is no secondary node and hence the node next in order is triggered to immediately start sensing and the sleep schedule is rescheduled.

D. Routing

In normal case, the data transmission from the primary nodes follow the hard coded routing path to reach the BS. In case of disaster leading to failure in the hard coded routing path the primary nodes follow DSR routing from that point with a goal to reach BS**.** Hence the techniques proposed when applied to WSN that uses a mesh topology for routing will increase the efficiency w.r.t., storage space, power utilization, and query processing. We propose a mathematical evaluation for our proposal in Section IV.

IV. MATHEMATICAL EVALUATION

In this Section we present the Mathematical evaluation of the proposed collaborative Sensing Scheme in comparison with the DIDS Scheme.

e_o = the maximum energy of all the nodes at time t=0

e_{th} = the minimum energy below which node ceases to function

E(n) = the energy left in a node at time t

Condition for the node t function: $E(n)-e_{th} > 0$ (1)

Let $\mathcal{E} = e_o - e_{th}$

Hence, $\mathcal{E} > 0$

Energy consumption of a node = energy consumption by battery + Energy needed to transmit data + Energy needed to receive data.

The energy consumption for control signal transmission and reception, data processing, data sensing is negligible and hence is not taken into calculation.

Energy consumption by battery = $K_1 e^{kt}$, where K_1, k are constants, and t is the time.

However as the sensor nodes used in the disaster prone areas have long life time, the slope of the energy consumption becomes constant from exponential and hence the equation becomes constant

Energy consumption by battery = $K_1 t$
Energy consumption on data reception = E_r
Energy consumption on data transmission = E_t
Total Energy consumption of a node = $K_1 t + E_r + E_t$
Energy of the node =
E(n) = Initial Energy – Energy consumption till time t
$= e_o - K_1 t + E_r + E_t$

Putting (1), $E(n) - e_{th} > 0$
$e_o - K_1 t + E_r + E_t - e_{th} > 0$
$e_o - e_{th} > K_1 t + E_r + E_t$
$\boldsymbol{\mathcal{E} > K_1 t + E_r + E_t}$ (2)

Let us consider that no of queries for each node is constant over a certain amount of time. Therefore, taking each query comes after a time slot tslot, the no of queries in time t is t/t_{slot}

As the main focus is on the intermediate nodes (storage nodes specially) as these are the nodes which have maximum energy consumption and hence when their energy is exhausted the data retrieval from that part of the network stops, our focus is thus to find out how long does the network functions. The longer the life time of these intermediate nodes, the more efficient and successful our proposal is

For the DIDS Scheme, It's a hierarchial structure and we shall consider the storage nodes energy consumption. Every storage node has its own data to transmit. It receives all the data from the sensing child nodes and storage child nodes. It transmits this data further.
$E_r = e_r.n_c.(t/t_{slot})$, where e_r is the energy consumption on every reception by a node and n_c is the number of child nodes

$E_t = e_t.n_c.(t/t_{slot}) + e_t.(t/t_{slot})$
$e_t.(n_c+1).(t/t_{slot})$, where e_t is the energy consumption on every transmission by a node and n_c is the number of child nodes.
Energy Consumption = $K_1 t + E_r + E_t$
⇨ $K_1 t + e_r.n_c.(t/t_{slot}) + e_t.(n_c+1).(t/t_{slot})$
⇨ $K_1 t + [e_r.n_c + e_t.(n_c+1)](t/t_{slot})$
⇨ $(K_1 + [e_r.n_c + e_t.(n_c+1)]/t_{slot}).t$
Putting (1) & (2),
$E(n) - e_{th} > 0$ (1)
⇨ $\mathcal{E} > K_1 t + E_r + E_t$ (2)
⇨ $\mathcal{E} > (K_1 + [e_r.n_c + e_t.(n_c+1)]/t_{slot}).t$
⇨ $\mathcal{E}/(K_1 + [e_r.n_c + e_t.(n_c+1)]/t_{slot}) > t$, this is the time for which the storage node lives

But the storage node has a backup with $E(n_1) > E(n_2)$, and hence $t_2 < t_1$.
Therefore condition of total life time of that network section =

$t_2 + t_1 < 2\mathcal{E}/(K_1 + [e_r.n_c + e_t.(n_c+1)]/t_{slot})$
$t_2 + t_1 < 2\mathcal{E}/(K_1 + [e_r.n_c + e_t.(n_c+1)]/ t_{slot})$
$t_2 + t_1 < 2\mathcal{E}.t_{slot}/[e_r.n_c + e_t.(n_c+1)]$ (3)

For the proposed Collaborative Sensing Scheme, As all the primary nodes are the intermediate connections between the secondary nodes of the cluster and next cluster nodes we shall take an assumption that the life time of a cluster depends on the actively functioning primary nodes. If all the primary nodes fail then the network section dies. It receives data from the other primary nodes in the cluster and transmits the aggregated data further. It receives all the data from the child clusters. It transmits this data further.
Taking the case of k = 8,
$E_r = 3e_r.(t/t_{slot}) + e_r.n_{cg}.(t/t_{slot})$
$e_r.(n_{cg}+3).(t/t_{slot})$, where e_r is the energy consumption on every reception by a node and n_{cg} is the number of child clusters
$E_t = e_t.n_{cg}.(t/t_{slot}) + e_t.(t/t_{slot})$
$e_t.(n_{cg}+1).(t/t_{slot})$, where e_t is the energy consumption on every transmission by a node and n_c is the number of child clusters.

Energy Consumption = $K_1 t + E_r + E_t$
⇨ $K_1 t + e_r.(n_{cg}+3).(t/t_{slot}) + e_t.(n_{cg}+1).(t/t_{slot})$
⇨ $K_1 t + [e_r.(n_{cg}+3) + e_t.(n_{cg}+1)](t/t_{slot})$
⇨ $(K_1 + [e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]/t_{slot}).t$

Putting equation 1 & equation 2,
$E(n) - e_{th} > 0$
⇨ $\mathcal{E} > K_1 t + E_r + E_t$ (2)
⇨ $\mathcal{E} > (K_1 + [e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]/t_{slot}).t$
⇨ $\mathcal{E}/(K_1 + [e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]/t_{slot}) > t_1$, this is the time for which the 1st primary node node lives
Total cluster life time = $t_1 + t_2 + t_3 + t_4 = t_{total}$
$t_{total} < 4\mathcal{E}/(K_1 + [e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]/ t_{slot})$
$t_{total} < 4\mathcal{E}.\ t_{slot} /[e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]$ (4)
For a network with large number of nodes (>100)
$n_{cg} \ll n_c$ (5)

Now comparing the lifetime of network sections for both DIDS and proposed collaborative sensing with equal number of nodes:
$2\mathcal{E}.t_{slot}/[e_r.n_c + e_t.(n_c+1)] < 4\mathcal{E}.\ t_{slot} /[e_r.(n_{cg}+3) + e_t.(n_{cg}+1)]$

Therefore it is mathematically proved that the proposed collaborative sensing has a multifold increase in the longevity of network life.

V. CONCLUSION

In this paper we have proposed an energy efficient data retrieval scheme which is best suited for disaster monitoring applications of WSN. The goal of our proposed approach is the availability of critical data during disasters. Our mathematical evaluation have proved the efficiency of our approach in terms of longevity of the network, load balancing and reduction of traffic due to frequent communication between neighbors to enhance network lifetime. The proposed collaborative sensing scheme focus essentially on the disaster monitoring and hence twin node sensing mode and cluster formation ensure the longevity of network and critical data retrieval. The sleep scheduling technique further lets the twin nodes to be in sleep for almost half of the time and thus energy efficiency is multifold. Our mathematical evaluation stands in support of our scheme.

REFERENCES

[1] Raghavendra, Cauligi S., Krishna M. Sivalingam, and Taieb Znati, eds. Wireless sensor networks. Springer, 2004.

[2] Benkhelifa, Imane, Nadia Nouali-Taboudjemat, and Samira Moussaoui. "Disaster management projects using wireless sensor networks: An overview." In *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 605-610. IEEE, 2014.

[3] Krishnamachari, L., Deborah Estrin, and Stephen Wicker. "The impact of data aggregation in wireless sensor networks." Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on. IEEE, 2002.

[4] Kimura, Naoto, and Shahram Latifi. "A survey on data compression in wireless sensor networks." Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on. Vol. 2. IEEE, 2005.

[5] Kavitha, Kayiram, Cheemakurthi Ravi Teja, and R. Gururaj. "Workload-Aware Tree Construction Algorithm for Wireless Sensor Networks." International Journal on Applications of Graph Theory in Wireless Ad Hoc Networks and Sensor Networks 4.1 (2012).

[6] Vuran, Mehmet C., Özgür B. Akan, and Ian F. Akyildiz. "Spatio-temporal correlation: theory and applications for wireless sensor networks." Computer Networks 45.3 (2004): 245-259.

[7] Seetharam, Anand, Abhishek Bhattacharyya, Mrinal K. Naskar, and Amitava Mukherjee. "Estimation of node density for an energy efficient deployment scheme in wireless sensor network." In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pp. 95-98. IEEE, 2008.

[8] Zach Shelby, Carlos Pomalaza-Raez, Heikki Karvonen and Jussi Haapola, "Energy Optimization in Multihop Wireless Embedded and Sensor Networks", International Journal of Wireless Information Networks, Springer Netherlands, January 2005, vol .12, no. 1, pp. 11-21.

[9] Liao, Wen-Hwa, and Mon-Shin Lin, "An energy-efficient sensor deployment scheme for wireless sensor networks." *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*. IEEE, 2011.

[10] Chakraborty, Suchetana, Sandip Chakraborty, Sukumar Nandi, and Sushanta Karmakar. "Exploring gradient in sensor deployment pattern for data gathering with sleep based energy saving." In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 1394-1399. IEEE, 2013.

[11] Bahrepour, Majid, Nirvana Meratnia, Mannes Poel, Zahra Taghikhaki, and Paul JM Havinga. "Distributed event detection in wireless sensor networks for disaster management." In *Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on*, pp. 507-512. IEEE, 2010.

[12] Cao, Qing, Tarek Abdelzaher, Tian He, and John Stankovic. "Towards optimal sleep-scheduling in sensor networks for rare-event detection." In *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 4. IEEE Press, 2005.

[13] Chao Gao, Bingwen Wang, Xiaoya Hu, Zhuguo Li and Hongliang Gao, 2011. An Efficient Index-based Data Storage Method for Wireless Sensor Networks. Information Technology Journal, 10: 1934-1941.