Scientific
Research

Energy and Throughput Optimized, Cluster Based Hierarchical Routing Algorithm for Heterogeneous Wireless Sensor Networks

Mahanth K Gowda, Kaushal K Shukla
Institute of Technology, Banaras Hindu University, Varanasi, India
E-mail: {*mahanth.gowda.cse*06, *kkshukla.cse*}*@itbhu.ac.in*
Received February 7, 2011; *revised March* 21, 2011; *accepted April* 7, 2011

Abstract

We propose a novel cluster based distributed routing algorithm in a generalized form for heterogeneous wireless sensor networks. Heterogeneity with respect to number/types of communication interfaces, their data rates and that with respect to energy dissipation model have been exploited for energy and throughput efficiency. The algorithm makes routing assignment optimized for throughput and energy and has a complexity of $\frac{N}{k} * \log N + k^2 \log k$ approximately, where N is the number of nodes and k is the number of cluster heads. Performance experiments confirm the effectiveness of throughput and energy optimizations. The importance of choosing an optimal cluster radius has been shown. The energy consumption in the network scales up well with respect to the network size.

Keywords: Routing Algorithm, Clustering, Heterogeneous Networks, WSN, Energy Efficiency

1. Introduction

A cluster based energy and throughput optimized routing algorithm for heterogeneous wireless sensor networks has been proposed here. Nodes in the network might be equipped with multiple radio/interfaces with diverse data rates. Nodes are assumed to possess different energies initially and they may dissipate energy at different rates depending upon the interface of communication being used.

The entire routing process comprises of three steps. Firstly, a value for cluster radius is computed for optimal energy consumption. Secondly, a set of clusters are formed within the network. Nodes with higher residual energy are more likely to become *Cluster Heads* (*CH*) here. Finally, intra-cluster route assignment is done through Linear Programming (LP). The LP combines both throughput and energy optimization factors in the objective function. With this route and Cluster assignments, nodes operate for a number of cycles or rounds before clusters are reformed and routes are re-assigned.

The entire algorithm is distributed and the complexity of route computation scales up well with network size. Various performance measurement experiments have

been conducted to study the behaviour of routing strategy. The results show that the energy optimizations and throughput optimizations perform well in optimizing their respective objectives. Also, the energy consumption in the network scales up well with respect to the network size.

2. Related Work

The work in [1] proposes a Linear Programming model for calculating upper bounds on network lifetime. LP equations model *CH* rotation within a same cluster for uniform consumption of energy resources. Once a network is divided into a set of clusters using genetic formulations, the authors use LP equations to make *CH* assignment and subsequent rotation. The use of genetic approach for the initial cluster formation makes the complexity of the algorithm, a polynomial with respect to network size. Thus, scalability issues arise when the size of the network is very large.

The work in [2] proposes a distributed approach for building energy efficient hierarchical clusters in constant time. The approach forms uniform clusters and makes use of residual energies of nodes while selecting *CH*s.

Nevertheless, there is a fair chance for nodes with lower energies to get grouped within a same cluster. This happens when nodes exhibit high degree of heterogeneity with respect to energies and having a high spatial density.

The work related with uniform cluster formation in [3] focuses on forming closely packed clusters with packing efficiency close to hexagonal close packing. Such cluster formations are effective in homogeneous networks with nodes having similar energy values, similar energy consumption functions and similar data loads. On a heterogeneous network, some non-uniform clusters biased towards energy rich nodes should have scope for formation.

The work on hierarchical cluster based routing in [4] proposes a distributed algorithm which first distributes the network into random clusters. Then, a set of high energy nodes within clusters are selected which serve as *CH*s. They also formulate a genetic algorithm for the initial cluster formation. The application of genetic approach makes the algorithm cost intensive for large networks.

The approach in [5] is for homogeneous networks. They formulate a set of Linear Programming equations to minimize the maximum cluster size in the network, since this is equivalent to minimizing energy consumption in a homogeneous network.

The work in [6], proposes a strategy to organize raw sensor nodes into clusters for energy efficient communication among nodes in the organized cluster. They calculate two parameters p (probability for a node becoming *CH* and set up *CH* election) and k (cluster hop radius). A set of mathematical equations are set up with these two as variables and these equations have been optimized to minimize total energy and thereby deriving optimal values for p and k. Based on these values they formulate a single level and multi-level cluster based routing algorithms. We use the results of this work to derive the optimal cluster radius which is required for formation of clusters.

The work in [7], formulates a set of Integer Programming equations for cluster assignment for optimal network lifetime. They then propose a heuristic algorithm for routing. However, the formulation doesn't consider nodes with multiple/radio interfaces. Also, the ILP (integer linear programming) and the heuristics use variables whose order is a polynomial with respect to the size of the network.

Dearth of published work in handling node heterogeneity in a generalized manner together with the idea of tuning the degree of conflicting optimization objectives of energy and throughput, have been the major sources of motivation to carry out the work.

3. Problem Description and Model

The nodes in the network are deployed randomly over a field in an adhoc mode. There is a base-station which offloads the sensory information collected by nodes to a wired network. The nodes sense information periodically and communicate amongst themselves to pass their data to the base-station.

Given the data load that nodes must sense periodically, the problem is to route the data so that the life-time of the network and throughput are optimized to a degree specified by the design parameter (managed by the network administrator). Nodes have multiple interfaces for communication with differing data rates, energy dissipation models.

4. The Routing Approach

The routing strategy involves three steps. Firstly, an optimal cluster radius based on analysis and approximations in [6] is chosen. Then, the entire network is divided into a set of clusters with nodes having higher residual energy having higher probability of becoming *CH*s. Then, a set of Linear Programming equations are solved within each cluster to assign routes for each node within a cluster towards the *CH*. These equations consider multiple radio/interfaces, their data rates, residual energy values of nodes, throughput of delivery etc into consideration while making route computations. These steps are explained in detail in the subsections to follow.

4.1. Optimal Cluster Radius Selection

Selection of an optimal cluster radius for minimum energy consumption is of prime importance. This has been done on the basis of work in [6]. The approach and how it has been adopted to our work has been briefed here.

Assume that nodes are distributed randomly with a density of λ in a spatial 2-dimensional square area. Now let N be a random variable representing the number of sensors in a square area of side $2a$ with mean λA ($A = 4a^2$). Given n nodes, if p is the probability that a node is a *CH*, then there will be np *CH*s in the area. Let D_i denote the length of the segment from node i (co-ordinates x_i, y_i) to the base-station which has been assumed to be at the centre of the square area without loss of generality. The expected value of D_i is given by

$$E\left[D_i \mid N = n\right] = \int_A \sqrt{x_i^2 + y_i^2} \left(\frac{1}{4a^2}\right) \mathrm{d}A = 0.765a$$

The other terms - E, N, n, N_v, λ_0, λ_1, C_1,

C_2, C_3, C - are treated as defined in [6].

Now, let the plane be divided into cells with each *CH* belonging to one (center) cell. **Figure 1** shows such an example. Let us assume that each non-*CH* joins the nearest cluster. Now the plane can be visualized as a set of exclusive zones of clusters. For this model, although there are minor changes in Equations (4), (6), (7) and (8) of [6], the final result of [6] still holds as shown below.

We use the square distance power consumption, *i.e.* the energy of transmission is proportional to the square of the distance transmitted. Now, if r is the average hop distance and $C1$ is the average energy consumption for transmitting equal data by all nodes in a cluster cell, then $C1$ is given by

$$E\left[C1\middle|N=n\right] = \frac{E\left[L_v\middle|N=n\right]}{r} * r^2 = rE\left[L_v\middle|N=n\right]$$

The first term in the product involved in the above equation $\dfrac{E\left[L_v\middle|N=n\right]}{r}$ is the expected number of hops and the second term r^2 is multiplied because the energy value is a square function of transmission distance.

Since there are np such cluster cells, the total energy spent $C2$, by all non - *CH*s in the square area is given by

$$E\left[C2\middle|N=n\right] = npE\left[C1\middle|N=n\right]$$

If the total energy spent by *CH*s to communicate the aggregated information to the processing center is denoted by $C3$, then,

$$E\left[C3\middle|N=n\right] = \frac{0.765npa}{r} * r^2 = 0.765npar$$

Note the modifications that have been done with computations of $C3$ and $C1$ in order to accommodate the square distance energy consumption.

Define C to be the total energy spent in the system. Then,

$$E\left[C\middle|N=n\right] = E\left[C2\middle|N=n\right] + E\left[C3\middle|N=n\right]$$
$$= \frac{npr(1-p)}{2p^{1.5}\sqrt{\lambda}} + 0.765npar$$

Removing the conditioning on *N* yields,

$$E\left[C\right] = E\left[E\left[C\middle|N=n\right]\right] = E\left[N\right]\left[\frac{(1-p)r}{2\sqrt{p\lambda}} + 0.765par\right]$$
$$= \lambda A\left[\frac{(1-p)r}{2\sqrt{p\lambda}} + 0.765par\right]$$

$E\left[C\right]$ is minimized by a value of p that is a solution of

$$cp^{\frac{3}{2}} - p - 1 = 0$$

where $c = 3.06a\sqrt{\lambda}$. This is same as Equation (9) in [6].

Accordingly, the real solution for p given by

$$p_{optimal} = \left(\frac{1}{3c} + \frac{2^{\frac{1}{3}}}{3c\left(2+27c^2+3\sqrt{3}c\sqrt{27c^2+4}\right)^{\frac{1}{3}}} + \frac{\left(2+27c^2+3\sqrt{3}c\sqrt{27c^2+4}\right)^{\frac{1}{3}}}{3c}\frac{1}{2^{\frac{1}{3}}}\right)^2$$

Here, we have $n * p_{optimal}$ clusters and let us assume that these clusters take shapes of squares of equal area as shown in **Figure 1**. Now, the optimal cluster radius $R_{cluster}$ is approximated as half diagonal distance of such a square.

$$R_{cluster} = \frac{2a}{\sqrt{2np_{optimal}}}$$

4.2. Cluster Formation Methodology

Once, an optimal cluster radius has been chosen, the network is divided into a set of exclusive clusters. The following steps outline the algorithm for cluster formation.

- **Sorting**. All nodes in the network are arranged in decreasing order of residual energy. Initially all nodes in the network are unclustered.
- **Cluster formation**. The node with highest residual energy among unclustered nodes is chosen as a *CH*. All nodes around it falling within the chosen cluster radius (which are neither *CH*s nor members of other clusters) become members of this cluster.
- **Termination**. The previous step is continued until each and every node in the network is either a *CH* or a member of a cluster.

Although the above algorithm runs in a polynomial order of the size of the network, it can be parallelized. The above algorithm has been presented for the purpose of better comprehension of the cluster assignment strategy. The distributed version of the above algorithm is given below.

- **Sorting**. Here, the nodes are sorted in decreasing order of residual energy in a distributed manner.

Figure 1. Sets of clusters with *CH*s at respective centres.

Initially a set of $k = np_{optimal}$ nodes are selected randomly (which are distributed uniformly across the network) which participate in sorting. Once the first round is complete, the k *CH*s of current round participate in sorting for cluster election for the next round.

The k nodes can perform distributed sorting by using *quick sort* algorithm presented in Subsection 9.4 of [8].

- **Cluster Head Selection**. After sorting, let *CH*s labelled as $CH_1, CH_2, \cdots, CH_k$ hold attributes for first, second $\cdots k^{th}$ set of $\frac{n}{k}$ nodes. If $E_{i,n}$ represents the residual energy of n_{th} node in CH_i, then $E_{i,a} > E_{i,b} \ \forall \ a < b$. Also, $E_{i,x} > E_{j,y} \ \forall i < j$. Based upon the chosen cluster radius R_{range}, a set of *CH*s which do not fall within a distance of R_{range} from each other are now selected from the sorted list.

The following is the algorithm at CH_i which has a list l_i of $\frac{n}{k}$ sorted nodes and $d_{m,n}$ is the distance between m^{th} and n^{th} nodes in l_i.

  for $m = 1$ to $l_i.size - 1$ **do**
    for $n = m + 1$ to $l_i.size$ **do**
      if $d_{m,n} \leq R_{cluster}$ **then**
        $l_i.remove(n)$
      end if
      end for
    end for
    if $l_i.size > 2k$ **then**
      modify l_i such that only the first $2k$ elements of the list are kept, rest are discarded

    end if
This is followed by the below algorithm.
  $j \leftarrow i$
  start:
  $it \leftarrow 0$
  if $j\%2 = 1$ **then**
    if $it > \lceil \log k \rceil - 1$ **then**
      goTo end
    end if
    $t \leftarrow i + 2^{it}$
    receive From (t, l_t) /*receive the list of node t*/
    append (l_i, l_t)
    for $m = 1$ to $l_i.size - 1$ **do**
      for $n = m + 1$ to $l_i.size$ **do**
        if $d_{m,n} \leq R_{cluster}$ **then**
          $l_i.remove(n)$
        end if
      end for
    end for
    if $l_i.size > 2k$ **then**
      modify l_i such that only the first $2k$ elements of the list are kept, rest are discarded
    end if
    $it \leftarrow it + 1$
    $j \leftarrow \dfrac{j+1}{2}$
  else
    send To $(i - 2^{it}, l_i)$/*send list to node $i - 2^{it}$*/
    goto end
  end if
  goTo start
  end:
  if $i = 1$ **then**
    send information to the nodes in l_1 to start *CH* advertisements within $R_{cluster}$.
  end if

The algorithm is analogous to parallel maximum-element determination among a series of numbers. At each step, each number is compared with its neighbour in the series. The winners (greater number in the comparison) go to the next round and the algorithm is terminated at the maximum element. In our algorithm, node-lists of *CH*s (CH_i and CH_j, where $i < j$) are compared and those nodes in l_j are discarded which come in the range of nodes of l_i. This is because nodes in l_i have higher residual energies than those in l_j. During this merging, CH_i proceeds with the next round taking a maximum of $2k$ nodes in its list. The algorithm terminates with CH_1 holding the potential candidates which form *Cluster Heads* for the next round.

- **Cluster Spawning**. These *CH*s now spawn their clusters by selecting the nodes that fall within a distance of R_{range} from them. If a node falls within R_{range} of two or more of the above *CH*s, then it will join the cluster of the *CH* with the largest residual energy among them. The following is the algorithm used by $node_i$.

> $highestResidualEnergy \leftarrow 0$
> $myCH \leftarrow i$
> **if** received a message from CH_1 for sending advertisements **then**
>> broadcast *CH* advertisement within $R_{cluster}$
>> $highestResidualEnergy \leftarrow myResidualEnergy$
> **end if**
> **while** $time \leq CHelectionTime$ **do**
>> Receive an advertisement *adv* from $node_j$
>> **if** $adv.residualEnergy >$ $highestResidualEnergy$ **then**
>>> $highestResidualEnergy \leftarrow$ $adv.residualEnergy$
>>> $myCH \leftarrow j$
>> **end if**
> **end while**
>> Each of the non-*CH*s acknowledge their respective *CH*s

CHelectionTime is a suitable predefined time for which the spawning takes place.

Figure 2 shows the initial network with nodes unclustered. Unclustered nodes are shown in green. **Figure 3** shows how clusters are built based upon residual energy values. A node among the unclustered nodes with highest residual energy spawns a new cluster. This continues until every node in the network is clustered. *CH*s are marked red while non *CH*s are shown yellow.

4.3. Intra Cluster Route Assignment through Linear Programming

The LP formulations presented here are for a single cluster. These equations are solved for each and every cluster separately.

The variables and parameters used in the LP formulation are tabulated in **Table 1** for quick reference. The column **T** indicates the type of the term - whether it is a variable or a parameter.

Optimization function:
minimize

$$\alpha * \sum_{i=1}^{N} FE_i + (1-\alpha)*linkSum$$

where, $0 \leq \alpha \leq 1$.

Constraints:
Equations representing constraints mostly dictate data
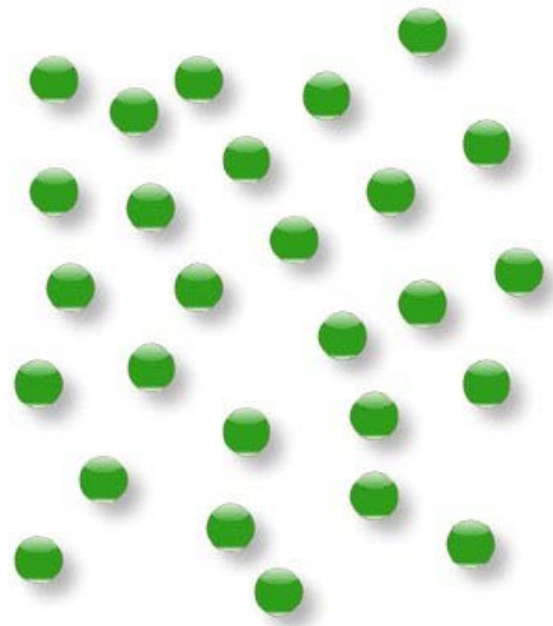

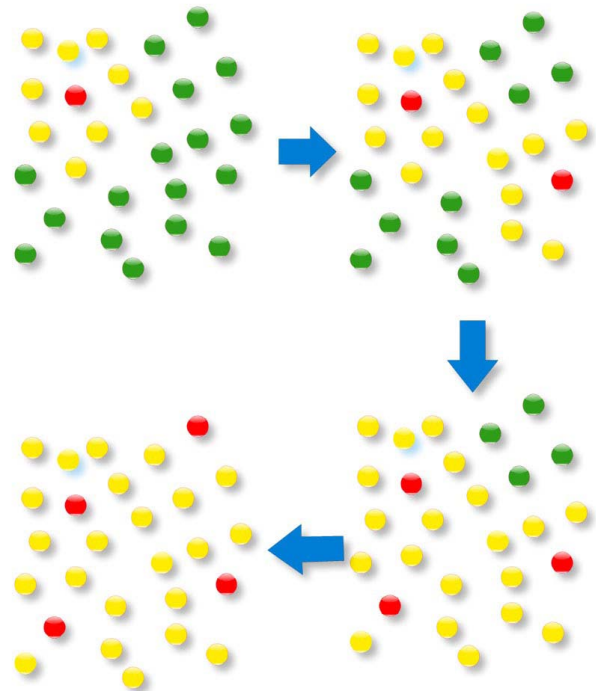**Figure 2. Initial unclustered network.**

Figure 3. Cluster formation based on residual energy.

balancing among nodes involved in routing. These equations take diverse aspects into account like multiple radio/ interfaces, energy, link data rate, throughput etc.

$$totald_i = b_i + recvd_i \qquad (1)$$

Total data received at a node should be equal to the sum of data forwarded by all other nodes to the node

along all interfaces.

$$\sum_{j=1,k=1}^{N,I} next_{ijk} = recvd_i \qquad (2)$$

A non-*CH* node should forward all of its received data and its own data to *CH* or to other nodes for onward transmission to the *CH*.

$$\sum_{i=1,k=1}^{N,I} next_{ijk} = totald_j \qquad (3)$$

$\forall j$ which is a non-*CH*.

$$next_{i,i,k} = 0 \qquad (4)$$

Energy for reception of data on a node is calculated by noting down all the data that is being transmitted to the node for onward transmission to *CH*.

$$E_{recp,i} = \sum_{j=1,k=1}^{N,I} a_{1k} * next_{i,j,k} \qquad (5)$$

Energy of transmission of data on a node is calculated by noting down the amounts of data, the node has to forward to other nodes, and their distances from the node. Square distance energy consumption model is used in which the transmission energy consumed is roughly proportional to the square of the distance of transmission as shown below.

$$E_{trans,i} = \sum_{k=1,,j=1}^{I,N} \left(a_{1k} + a_{2k} * d_{ij}^2 \right) * next_{jik} \qquad (6)$$

Fraction of energy consumed at a node is calculated by taking a ratio of total transmission energy and reception energy consumed at a node and its initial energy.

$$FE_i = \frac{E_{recp,i} + E_{trans,i}}{E_{initial,i}} \qquad (7)$$

Sum of transmission costs at each link is calculated as below. The *linkSum* has to be minimized for optimal throughput.

$$linkSum = \sum_{i=1,j=1,k=1}^{N,N,I} \frac{next_{ijk}}{rate_k} \qquad (8)$$

All data in the cluster should reach the *CH*.

$$recvd_{ch} = \sum_{j=1}^{N} b_j - b_{ch} \qquad (9)$$

where *ch* is the **Cluster Head**.

Some nodes might not have some communication interfaces.

$$next_{ijk} = 0 \qquad (10)$$

$\forall$ pairs (i,j) where one of the members of the pair does not have communication interface *k*.

4.4. Routing Algorithm on Nodes after Route Assignment

The LP is run on *CH*s. Based on the values of $next_{ijk}$, *CH*s create a variable *nextlist* on each node of their respective clusters. This contains a list of nodes to which the given node has to transmit data along with the the amount of data and the interface on which the communication has to be made.

```
radios ← on
dataReceived_j ← 0
while  dataReceived_j < recvd_j  do
    dataReceived_j ← dataReceived_j + ReceiveData()
end while
while  nextList ≠ NULL  do
    SendData(nextList−>i, nextList−>k, nextList−>
    data) //(send (nextList−>data) to node
    (nextList−>i) using interface (nextList−>k))
    nextList ← nextList−>next
end while
radios ← off
```

Once the data reaches *CH*s, it is routed amongst themselves towards the base-station using a shortest path based algorithm.

5. Complexity Analysis

Overall complexity of the cluster formation algorithm is determined by taking the individual complexities of the following phases. Assume *k CH* s and *n* nodes.

- **Sorting**

$$\theta\left(\frac{n}{k}\log\frac{n}{k}\right) + \theta\left(\frac{n}{k}\log k\right) + \theta\left(\log^2 k\right)$$

- ***CH* selection and spawning** As seen in Subsection 4.2, the algorithm for initial selection of nodes which send advertisements consists of $\log k$ steps. At each of these steps, discarding a lower energy node which falls within a distance of $R_{cluster}$ from a higher energy node takes at most $\theta(k^2)$ time. Thus, the overall complexity is given by

$$\theta(k^2 \log k)$$

- **Route assignment with Linear Programming**: GNU's Linear Programming toolkit [9] uses primal and dual simplex methods to solve LP equations in a polynomial time. Thus, the order for this phase can be written as:

$$\theta\left(\frac{n}{k}\right)^p$$

The value of *k* depends on cluster density as shown

earlier in Subsection 4.1. In any case, the distributed approach brings down the complexity of the algorithm.

6. Advantages of the Approach

Some of the advantages of our approach are listed below:

- Since nodes have a precomputed routing assignment via LP solution, most of the non-*CH*s can turn their radios on/off appropriately to conserve energy.
- Non-*CH*s are exempted from the overhead of maintaining routing state. Thus, they can run with constrained memory.
- Reconfiguration and re-formation of clusters periodically ensures uniform consumption of energy resources across *CH*s. LP based solution ensures uniform energy consumption within clusters.
- Generalized approach taking care of multiple node heterogeneity like multiple radio/interfaces, energy values, data transmission rates etc in the formulation.
- A tunable design parameter α is provided to the end user to specify the degree of performance of the algorithm with respect to conflicting optimization requirements - throughput vs energy.

7. Results

It has to be noted that our research work with distributed clustering algorithm is very much different from other published work. It is a novel approach treating heterogeneity with respect to multiple node interfaces, data transfer rates, initial energy values and power consumption models. The tunable design parameter α can be used to vary the degree of performance of the algorithm with respect to throughput optimizations and energy optimizations. Thus, the performance evaluation done in this section evaluate the algorithm with respect to its own parameter settings.

The results show that the energy optimizations and throughput optimizations perform well in optimizing their respective objectives. Also, the energy consumption in the network scales up well with respect to the network size.

Implementation of the algorithms discussed in the document have been done on ns3 wireless networks. [10].

7.1. Some Definitions

- **Network Lifetime**: The network is assumed to be dead when atleast 1% of nodes in the network have depleted their energy completely. Accordingly, the **Network Lifetime** has been defined as the diffe-

rence between the point of time at which 1% nodes deplete their energy and the point of time at which the network starts its operation.
- **Datapercycle**: Nodes in the network operate in cycles. During each cycle, a node has to transmit some information to its *CH*. The amount of data per cycle which a node is supposed to sense and transmit is termed as Datapercycle.
- **Cycle Time**: The total time taken by all nodes in the network to finish with the transmission of their Datapercycle to their *CH*s.

The network operates in rounds or cycles. During each cycle , every node has to transmit its *Datapercycle* to its *CH* for the onward transmission to the **Base Station**. The energy spent during such transmissions has been accounted for analysis in various experiments. After each cycle, clusters are reformed and routes are reassigned.

7.2. Network Characteristics

Network characteristics have been summarized in **Table 2**.

Nodes for simulation were generated randomly on a square area and were assigned initial energies randomly in the specified range. Several plots (exp 1, exp 2, exp 3) have been made for the same experiment by initializing the seed value of the random node generator with different values.

7.3. Route Assignment

Consider the following cluster of nodes – 318, 280, 298, 299, 319, 320, 321, 335, 336, 337, 349 with node 318 as the *CH*. Other parameters defined for these nodes are shown in **Table 3**. In the table, id, x_i, y_i stand for identities of nodes, their x-coordinates and y-coordinates respectively. Also, *intf* holds the number of interfaces a node can have while b_i and $E_{initial,i}$ have been defined in **Table 1**.

Solving LP for the above cluster yields values for $next_{ijk}$ and $recvd_i$ which have been tabulated in **Tables 4** and **5**. Values for other possible $next_{ijk}$'s and $recvd_i$'s which are not listed in these tables were found to be 0.

Since the value of the tunable design parameter α that was used in this experiment was 0.0001, almost all of the nodes use the interface q for communication which was available on all of them.

Now, if the algorithm in Subsection 4.4 is applied to node 336, it receives 34 units of data (25 and 9 respectively from nodes 320 from 337 along interface q). Once the data is received, the node transmits 44 (34 received plus 10 of its own) units of data to node 319 on

Table 1. Terms and definitions.

Term	Definition	T
d_{ij}	distance between i th and j th nodes	p
b_i	Amount of data to be sensed and transmitted to basestation by i th node during each round	p
$recvd_i$	Total data received by node i from other nodes during a round for forwarding to CH	v
$totald_i$	sum of $recvd_i$ and b_i	v
$next_{i,j,k}$	Amount of data to be forwarded by node j to node i on interface k during a round	v
$a_{1,k}$	A parameter of interface k's energy function	p
$a_{2,k}$	A parameter of interface k's energy function	p
$E_{recp,i}$	Total energy spent for reception by node i during a round	v
$E_{trans,i}$	Total Energy spent for transmission by node i during a round	v
FE_i	Fraction of energy consumed on node i during the operation of a round	v
$E_{initial,i}$	Initial energy of node i before the start of the round for which optimal assignment is being done.	p
$rate_k$	rate of data transmission of k th interface	p
$linkSum$	sum of ratios of data transferred along links by nodes and the data transfer rate of the corresponding link	v
α	tunable design parameter which can be set values between 0 and 1	p

Table 2. Network characteristics.

Network Parameter	Value
Number of nodes	ranging from 270 – 1020 over all experiments
Number of interfaces per node	All have $interface_1$ and 40% have $interface_2$ also
datarate $interface_1$	1 Mbps
datarate $interface_2$	2 Mbps
datapercycle	1-2 KB
a_1 for $interface_1$	4
a_2 for $interface_1$	1
a_1 for $interface_2$	8
a_2 for $interface_2$	2
Area of deployment	10^6 m^2
Initial Energy of nodes	300 - 1000 units

Table 3. Illustration of routing assignment.

id	x_i	y_i	b_i	intf	E_i
318	568	690	10	q,r	520
280	693	614	8	q	364
298	510	631	8	q	457
299	560	646	7	q,r	327
319	622	676	8	q	441
320	688	697	7	q	347
321	710	695	10	q,r	202
335	629	731	10	q	426
336	637	708	10	q	75
337	690	723	9	q	308
349	653	767	6	q,r	267

interface q.

7.4. Plot of Network Lifetime versus α

A plot of **Network Lifetime** on *y*-axis versus α (the design parameter introduced in Subsection 4.3) on *x*-axis for two random experiments is shown in **Figure 4**. The network lifetime has been measured in terms of number of **NS3 Simulation seconds**. A high value of α makes the lifetime optimization part of the optimization function in LP formulation to dominate, and a low value makes the throughput optimization part to dominate. The results of the experiment which depict this behaviour is shown in the graph.

7.5. Plot of Cycle Time versus α

A plot of **Cycle Time** on *y*-axis versus α on *x*-axis for

three random experiments is shown in **Figure 5**. Cycle Times have been measured in terms of number of **NS3 Simulation seconds**. A low value of α makes throughput optimization part to dominate and consequently the cycle time is low for such values. On the other hand, for high values of α, the cycle time goes up. The results are depicted in the graph.

7.6. Plot of Energy Consumed versus Node Density

A plot of total energy consumed in the network versus Node Density for 1 cycle of operation (at optimal cluster radius as derived in Subsection 4.1) is done in **Figure 6**.

Table 4. LP solution for *next*$_{ijk}$.

i	j	k	*next*$_{ijk}$
388	299	q	15
318	319	q	68
299	298	q	8
319	335	q	16
319	336	q	44
320	280	q	8
320	321	q	10
335	349	q	6
336	320	q	25
336	337	q	9

Table 5. LP solution for *recvd*$_i$.

i	*recvd*$_i$
318	83
299	8
319	60
320	18
335	6
336	34

Plots for three random experiments have been shown. Energy values calculated as per Equations (5) and (6) have been scaled down by a constant value of 300 000. Node density has been shown in terms of number of nodes per 10^6 m^2 square area. The plot shows that the total energy consumed in the network scales up well with respect to the network size.

7.7. Plot of Energy Consumed vs Cluster Radius for Different Node Densities

Figure 7 shows the plot of **energy consumed** vs **cluster radius** for different **node densities**. Node densities have been shown as number of nodes per 10^6 m^2 for each of the different curves. Curves have expected shapes with minimas. However there is a discrepancy between theoretical optimal cluster radius and actual optimal cluster radius which can be explained on the basis of the following reasons.

- The theoretical analysis and derivation assumes uniform clusters, whereas actual clusters formed are non uniform with variable number of nodes.
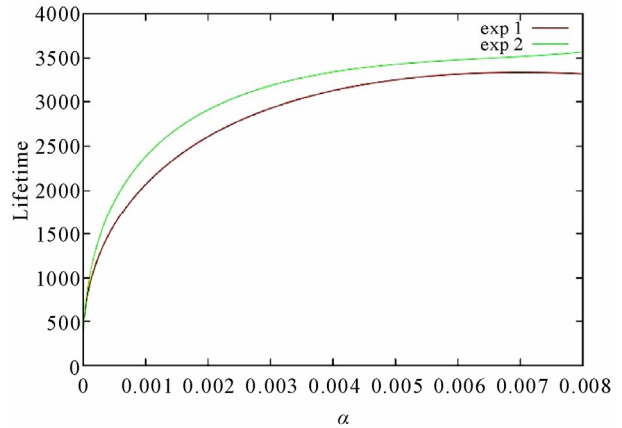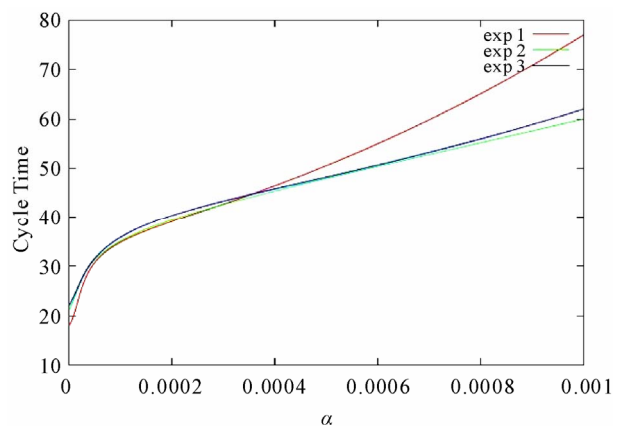

Figure 4. Plot of lifetime vs α.
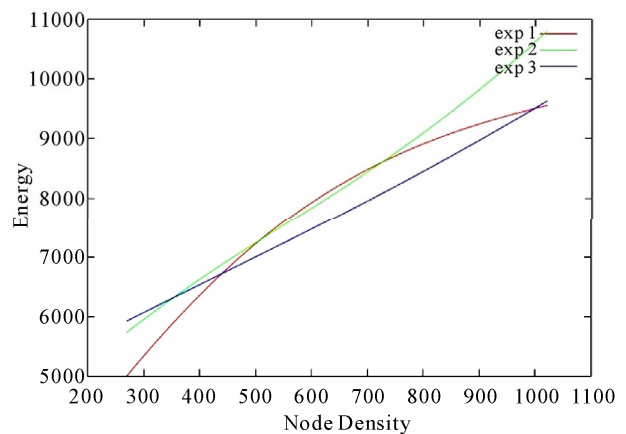

Figure 5. Plot of cycletime vs α.

Figure 6. Plot of energy vs node density.

- The theoretical analysis assumes square clusters, whereas the actual clusters which are formed can have any shape.

Results of experiments show that 1.5 times the theoretical cluster radius should be a good approximation for the actual optimal cluster radius for our network.
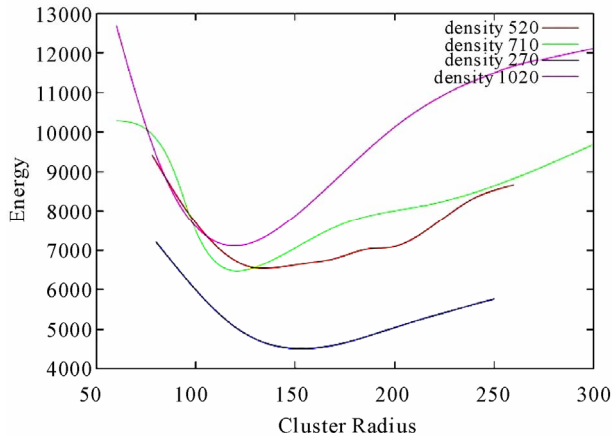
Figure 7. Plot of energy consumed vs cluster radius for different node densities.

8. Conclusions and Future Works

In this work, we have discussed the design, implementation and results of a hierarchical throughput and energy optimized, cluster based routing algorithm for heterogeneous wireless sensor networks. Firstly, the algorithm selects an optimal value for cluster radius, based on random variable analysis and approximations. Then, cluster assignment based on residual energies of nodes are made in a distributed manner using the cluster radius computed. Finally, Linear Programming equations are used for optimal intra-cluster routing assignment. The formulation considers multiple radios, interfaces, energy values, throughput of delivery etc of nodes into account. Also, there is a tunable design parameter α which can be varied depending upon the requirements of performance optimizations - throughput vs energy.

Various performance measurement experiments have been conducted and the results have been reported. These results show that the energy optimizations and throughput optimizations perform well in optimizing their respective objectives. The significance of choosing an optimal cluster radius has been shown. Also, the energy consumption in the network scales up well with respect to the network size.

In the near future, we would like work on the following aspects:

- Design of a message prioritization scheme such that throughput optimized paths are reserved for prior most messages while, ordinary ones can take energy optimized paths.
- Extension of the algorithm to support a generalized multi-level cluster hierarchy.
- Invention of suitable techniques for handling security and privacy.

9. References

[1] M. Qin and R. Zimmermann, "Studying Upper Bounds on Sensor Network Lifetime by Genetic Clustering," *Distributed Computing in Sensor Systems*, Vol. 3560, 2005, p. 465. doi:10.1007/11502593_40

[2] P. Ding, J. Holliday and A. Celik, "Distributed Energy-Effcient Hierarchical Clustering for Wireless Sensor Networks," *Distributed Computing in Sensor Systems*, Vol. 3560, 2005, pp. 466-467. doi:10.1007/11502593_25

[3] H. Chan and A. Perrig, "Ace: An Emergent Algorithm for Highly Uniform Cluster Formation," *Wireless Sensor Networks*, Vol. 2920, 2004, pp. 154-171. doi:10.1007/978-3-540-24606-0_11

[4] S. Hussain and A. W. Matin, "Hierarchical Cluster-Based Routing in Wireless Sensor Networks," *The 5th International Conference on Information Processing in Sensor Networks*, Nashville, 19-21 April 2006.

[5] W. Z. Wang, W.-Z. Song, X.-Y. Li and M.-N. Kosha, "Distributed Computing in Sensor Systems: Third IEEE International Conference," Springer, Berlin, 2007.

[6] S. Bandyopadhyay and E. J. Coyle, "An Energy Effcient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *INFOCOM* 2003 *The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, 30 March - 3 April 2003, pp. 1713-1723.

[7] A. Bari, A. Jaekel and S. Bandyopadhyay, "Clustering Strategies for Improving the Lifetime of Two-Tiered Sensor Networks," *Computer Communications*, Vol. 31, No. 14, 2008, pp. 3451-3459. doi:10.1016/j.comcom.2008.05.038

[8] A. Grama, A. Gupta, G. Karypis and V. Kumar, "Introduction to Parallel Computing," Addison-Wesley Longman Publishing Co., Boston, 2003.

[9] GNU Linear Programming Kit, 2011. http://www.gnu.org/software/glpk

[10] The ns-3 Network Simulator, 2011. www.nsnam.org