

INTERFACE ACTIVATION IN MULTI – RADIO
WIRELESS NETWORKS

By

VIJAYARAJA SAHADEVAN

Bachelor of Science
Anna University
Chennai, India
2005

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
In partial fulfillment of
the requirement for
the Degree of
MASTER OF SCIENCE
July 2008

INTERFACE ACTIVATION IN MULTI - RADIO
WIRELESS NETWORKS

Thesis Approved:

Dr. Venkatesh Sarangan

Thesis Adviser

Dr. Johnson P. Thomas

Dr. Nohpill Park

Dr. A. Gordon Emslie

Dean of the Graduate College

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	01
II. PROBLEM DESCRIPTION.....	03
Shortest Energy Path and minimizing hop count.....	03
Minimizing Number of Turned on radios.....	04
Proposed Heuristic.....	05
III. PROBLEM DEFINITION.....	07
IV. RELATED WORK.....	08
V. SOLUTION.....	09
Assumptions.....	09
Initialization.....	09
Algorithm.....	10
VI. RESULTS.....	17
Simulation Settings.....	17
Performance Metric.....	17
Methods Compared.....	18
Experiment 1.....	18
Experiment 2.....	19
Experiment 3.....	20
SUMMARY.....	22
REFERENCES.....	23

LIST OF FIGURES

Figure		Page
1.	Sample Network.....	03
2.	Shortest Energy path.....	04
3.	Minimizing number of turned on HP radios.....	05
4.	Proposed Heuristic.....	06
5.	Expected Connected Network.....	10
6.	HP radios turned on.....	19
7.	Average Energy Path.....	20
8.	Average Hop count.....	21

NOMENCLATURE

Base – The main station where all nodes transfer their data

BFS – Breadth First Search

E – Edge between two nodes

G - Graph

HP – High Power

NP – Non deterministic Polynomial time

Sink – It is same as the Base, the names are used interchangeably

V - Set of sensor nodes in the graph G

CHAPTER 1

INTRODUCTION

A wireless sensor network consists of autonomous devices spatially distributed using sensors to monitor the physical or environmental conditions co-operatively, for example pressure, sound, temperature, etc[1]. at different locations, for e.g. forest monitoring, disaster management, space exploration, [2] Environment monitoring in coal mines is an important application of wireless sensor networks [3], Habitat monitoring of real life creatures [4]. In the recent past, wireless networks have found their way into a wide variety of applications and systems with vastly varying requirements and characteristics. Each sensor node has a micro controller, a wireless communication device and an energy source. The energy source is vital for the node to functionally be active. Conserving energy in every node is helpful for the network to stay connected and send useful information to the base station for longer time. A sensor network is a hot research topic currently trying to improve and utilize the sensor network efficiently.

Every node has a limited amount of energy source which is vital for the network to stay connected. To maximize utilization of the energy source in each node is a research issue in sensor networks. Making the energy source to be used efficiently allows the nodes to be active for a longer time, which means that the network can stay connected for a longer time. The goals we discuss in this thesis are to maximize the lifetime of the network and to minimize the maximum latency, i.e. to reduce the number of hops. Assume all the nodes have a single radio. To find the minimum energy consumed along the path the shortest energy cost path can be used to find it, and as it is a single radio all radios have to be turned on by default so the total energy cost on the network has to be the summation of energy cost at all nodes. Also we need to avoid over using a single sensor node in the network because there is a greater chance of draining the node's energy and as an effect the network might be disconnected based on the topology of the network.

We have advances in technologies that support multiple radios with different capabilities and interfaces on a single sensor node platform [5]. Multiple radios allow the network to be connected easily by turning on all the highest frequency radios. The issue that we might encounter is high power radios consume more energy and the network might not be active for the most time possible. Turning on all the low power nodes reduces the network energy, however there are chances that the network might not be connected, and the main purpose is not achieved here. In order to achieve the main goals discussed above, we can find the shortest energy path for each node. The shortest energy path means less energy is consumed along the path to transfer data to the base. If we apply the shortest energy path for each node to the base, the overall energy in the network goes up as we might need to turn on more of the high power radios in the nodes which results in increasing the overall network energy. The second issue in the multi radio sensor nodes to reduce the energy consumption is to turn on as few radios as possible. This is a kind of group Steiner problem, where the ability to add new nodes is denied, which is studied and known to be a NP-hard problem [6]. This might be the best solution for reducing the overall network energy and also reducing the energy path from each node to the base. This might be hard to find for very large sensor networks. The third issue is to avoid over using a single node in the sensor network so that we avoid the sensor network to be disconnected due to few nodes being inactive. If we can merge all the three problems and find the solution it gives the best solution for a sensor network to stay connected for a longer time and send useful information. We are neglecting the third issue to make the problem less complicated.

We are trying to propose a heuristic to solve this problem to reduce the overall energy consumption in multi radio sensor network and minimize the maximum latency for each node to the base. In other words we try to find a way to activate the minimum number of radios and at the same time route the shortest path from the node to the base. When network nodes have multiple radios, the shortest path algorithm does not perform well [7] i.e. reduce the hop count from the node to the base station. Both these problems are interdependent, and trying to reduce one results in the increase of the other. The heuristic must be able to find the interface activation in a given time and at the same time try to minimize the energy consumption as much as possible.

CHAPTER 2

PROBLEM DESCRIPTION

Let each nodes have two different power radios one the low power and the other is the high power radio, two frequency radios can be extended to n frequency radios at the end. Let us assume a graph $G(V,E)$ where

V is the set of sensor Nodes

E is the path or edges between the nodes. As the nodes have multi radio each node can be connected by more than an edge i.e. each node is connected with all the available radios in range. For example,

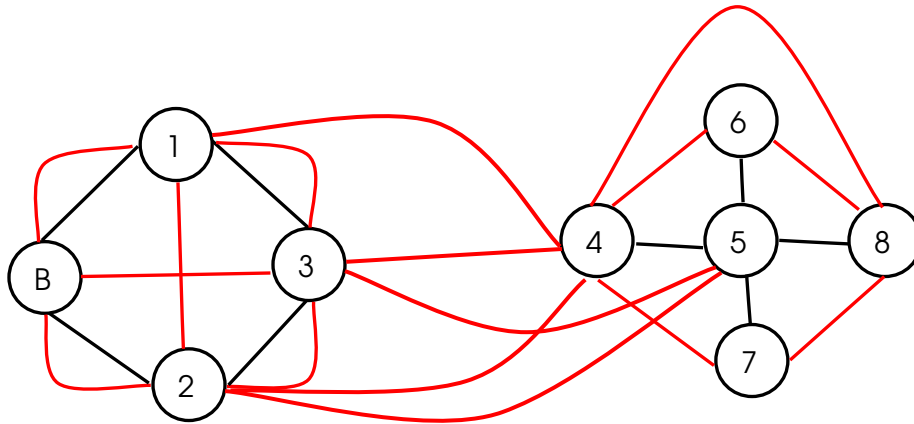


Fig 1 – Sample Network

B - Base station or Sink
— - Lower power edges
— - High Power edges

In the above graph all nodes are identified using their numbers. Let us see the resulting graphs for the shortest path route and minimum total energy consumed in the graph. We consider the two probable solutions.

Shortest Energy Path and minimizing hop count

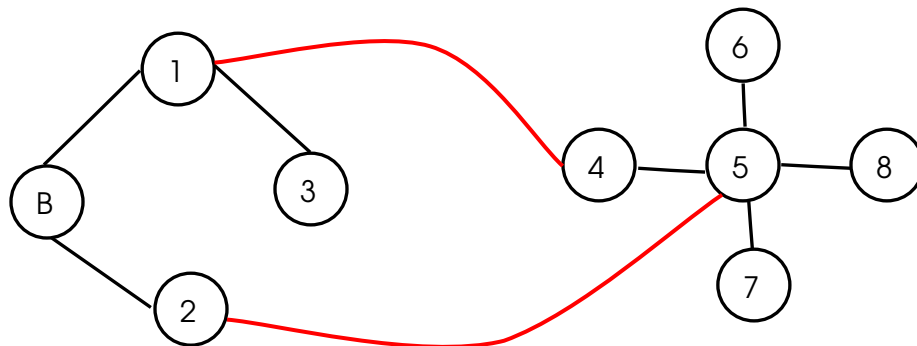


Fig 2 – Shortest energy path and minimizing hop count

In the above graph for shortest path with respect to power consumed along the path we can see that a total of four high power interfaces are turned on at 1, 2, 4 and 5 which increases the energy consumption of the overall network. In order to find the shortest energy path we eliminate the high power radio connections if the two nodes are already connected with low power radio. It is not an issue with the next case as minimum spanning tree eliminates the high power radios on its own.

Minimizing the number of turned on HP radios

We try to minimize the number of turned on HP radios to reduce the overall energy consumed in the network, which is one of our main goals. We consider Minimum Spanning Tree to get the minimal number of high power radios turned on.

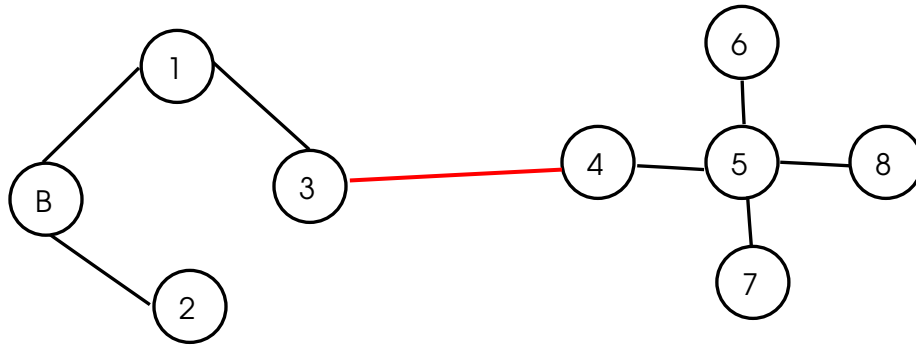


Fig 3 – Minimizing number of turned on HP radios

Minimizing the number of radios turned on can be found by implementing a Minimum Spanning Tree algorithm which first finds all the low power radios and then tries to turn on few high power radios to make the network connected. The difference between the Shortest Path and Minimum Spanning tree is that for example consider node 6, in shortest path it takes 3 hops for it to send data to the base where as here it requires 5 hops to take the data to the base. However the overall energy consumed in the network has been considerably reduced as 2 high power nodes are turned off here. So there is always a trade-off between the Shortest Energy path and minimizing the number of radios turned on.

Proposed Heuristic

Our heuristic should find a resulting graph similar to Fig 4., which tries to reduce the total energy consumed in the Network and at the same time tries to reduce the number of hops of each nodes to the sink. As we see from the below network, duplicate interface connections have been eliminated, and the network is better connected, accomplishing our above goals compared with the other two algorithms.

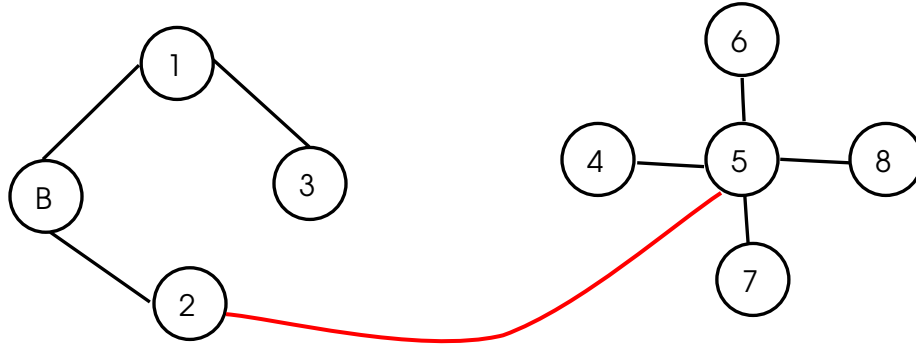


Fig 4 – Proposed Heuristic

We try to find a solution as the above Fig and improve it to be as close the functionality Shortest Path and Minimum Spanning Tree as possible and we analyze the differences in energy consumption and energy consumed along the path. From the figure we can see that our heuristic tries to get the balance between the number of HP radios turned on and at the same time to reduce the hop count or energy consumed along the paths which are inversely proportional. If we try to reduce one factor it results in the increase of the other.

We also try to solve the heuristic in a reverse manner, i.e. instead of finding the unfound islands from the found island what might happen if the unfound islands try to explore to a found island. We have given an algorithm in Heuristic 2 and also its results along with the others to compare it with each other.

CHAPTER 3

PROBLEM DEFINITION

Let v_1, v_2, \dots, v_n be the nodes present in a wireless sensor network. Each node v_i is equipped with N radios r_1, r_2, \dots, r_N . Let R_k and P_k respectively be the range and power consumption of radio k . Let $I_{i,k}$ be an indicator variable which denotes the on/off status of radio k in sensor node v_i , i.e. $I_{i,k}=1$ if radio k is turned on in node v_i , and is 0 if it is turned off. Two nodes v_i and v_j are said to have an edge of type k between them iff $d(v_i, v_j) \leq R_k$ and $I_{i,k} = I_{j,k} = 1$, where $d(v_i, v_j)$ denotes the geographical distance between the nodes v_i and v_j . Thus two nodes can have a maximum of K edges between them. A *path* is said to exist between two nodes v_i and v_j if a sequence of distinct nodes $v_i, v_a, v_b, \dots, v_m, v_j$ can be found such that any two adjacent nodes in the sequence have an edge of some type between them. With these notations, the problem we aim to solve can be defined as follows.

1) A path exists between v_i and $v_j \forall i, j \in \{1, 2, \dots, n\}$, and

2) $\sum_{i=1}^n \sum_{k=1}^K I_{i,k} \cdot P_k$ is minimized.

In other words, the objective is to make the network connected while minimizing the energy spent in all the active radios. The problem is briefly described in the chapter 5.

CHAPTER 4

RELATED WORK

Not much work has been reported in the literature on multi-radios for wireless sensor network in Multi – radio wireless sensor networks. Earlier works on multi-radio systems have focused on transmission scheduling [8], selectively waking up nodes [9], hierarchical power management [10], exploiting the capacity of the high capacity radio in a tiered architecture, and using the second radio as a paging and control channel for discovering resources and neighbors and mobility support [11], [12]. There are existing works where researchers assumed that all radios had to be turned on in order for the network to stay connected, and there are existing works where high power radios are turned on only when required for the network to stay connected. There might be a few networks which stay disconnected if these high power nodes are turned off. The network cannot be expected to be disconnected because these high power radios are turned off either it cannot turn on all the high power radios all the time which drains the life of the node quickly and the networks go disconnected when the battery dies.

There are works which deal with routing in dual-radio networks where a sensor node turn on its high power radio and leaves it on until it finds some other node with its high power radio on to be in range or selectively wakes up high power radios in nodes along the path when ever required to transfer data[9]. Selective wake up might not be advantageous in emergency applications where we might need to send data during emergencies immediately to the base or sink without any delay.

CHAPTER 5

SOLUTION

Assumption

We assume that all sensor nodes have two radios of different power which can later be extended to K power radios.

Initialization

All nodes in the Network are initialized to be unfound and also their precedence is initialized to be blank and the distance from to the sink is initialized to zero. Connect all the low power radios in the nodes and eliminate the high power radios. Connecting all low power radios forms islands or groups of nodes that can be connected through each other in that island. From the above example B, 1, 2, and 3 form a group or island and the rest forms the other island. Once two nodes are connected to each other through low power nodes then the high power nodes between these two islands can be eliminated. After finding all the groups in the graph, name them and find the centers of each group by finding the node whose maximum depth is minimal in the group. We need the groups and centers to start with our heuristic. Also the center for the island which contains the base is the base itself.

Algorithm:

Assume all the groups and centers of each group are found. Now we need to activate the high power nodes to make the groups connected. Start from the sink and check if the sink can be connected to any of the island's centers, connect if it has a high power radio. This should probably be a good solution to reduce the energy consumption on the network. In case if the sink (center) cannot be connected to the center of group two, try to find the node nearest to the center that can be connected from the sink. And we need to continue this process for all the nodes until all the groups are connected to make the network connected. Consider the example below when is found from the Fig 1.

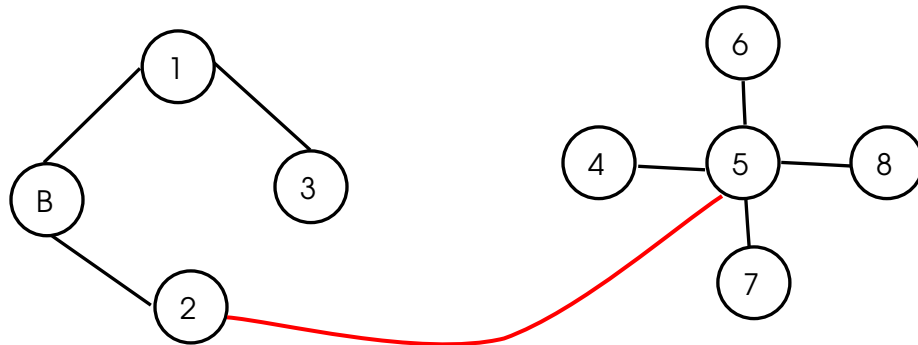


Fig 5 – Expected connected network

We first do a BFS from base B, and find node 1 and node 2, which in turn finds node 3. The center of the second group is 5. Try to connect B with 5 and there is no high power radio so they cannot be connected. Then we try to connect B with the nodes in group 2 which are 1 low radio hop from the center. B cannot connect to any of the four nodes. Then we consider node 1 and 2 and try them to connect to the centers. Node 2 has a high power radio connection to the center 5, and 1 can be connected to 4. We need to choose the one that connects to the center as it reduce the local hop count in group two. Then a BFS with low power radios is done starting from 5 as it has been found. Then group 2 is completely found and the whole network is connected. Thus we get a balanced solution between shortest path and minimum spanning tree.

The whole algorithm needs to be done in the following three steps:

1. Find the islands in the network.
2. Find the center of each island.
3. Interface activation (Heuristic I).

Find the islands in the network: BFS can be used to find the islands in the network. Initialize all nodes to be unfound. Pick any unfound node in the island and do a BFS using low power radios only. This tree forms one island. Now do BFS similarly using only low power radios until all the nodes in the network are found, i.e. all nodes should be contained in an island in the network. Name all the islands from 1 to I. The complexity to find the islands in the network is $O(I(J_{max} + E_{max}))$

Where,

J_{max} – number of nodes in the island

E_{max} – number of edges in the island

I – number of islands in the network.

Now,

Lets assume there is only one island then I becomes 1, J_{max} becomes V and E_{max} becomes E. So, the complexity becomes $O(V+E)$.

Finding the center of the island: The center in the island is the one from which all nodes in the island can be reached relatively quicker compared to the other nodes. Do a BFS on all nodes in the island and pick the node whose maximum depth in the tree is minimal. Similarly centers of all islands need to be found. The center for the island which contains the Sink is the sink itself because all data needs to be sent to the sink. Assume we have only one island. Now the complexity for doing a BFS on a single node in the island is $O(V+E)$. We need to do a BFS on every node so the complexity turns out to be $O(V(V+E))$, i.e. $O(V+E)$ is done V times. Now we need to find the node with minimal depth whose complexity is $O(V)$. Summing up, we get the complexity to find the centers to be in the order of $O(V^2)$.

Interface Activation (Heuristic I): The pseudo-code of the designed algorithm is given below.

1. **For** Each U in V[G]
2. **Do**
3. Node[U].color <- White


```

4.         Node[U].precede <- Negative
5.         Node[U].distance <- infinite
6.         Color2[U] <- white
7.     For Each I in Island[G]
8.         Island[I].color <- white
9.     Q <- Sink
10.    T <- Sink
11.    Node[Sink].color <- Gray
12.    Node[Sink].distance <- Zero
13.    Node[Sink].color <- Gray
14.    Node[Sink].precede <- Negative
15.    while T ≠ NULL
16.        For each W in Island[Sink]
17.            Do Relax of T.front()
18.            Update color, precede and distance as needed.
19.            Q <- all relaxed nodes of T.front()
20.            T <- all relax nodes of T.front()
21.            T.pop()
22.    while Q ≠ NULL
23.        For all I in Island[G]
24.            If color[I] == white
25.                S <- I.center
26.                I.center <- Gray
27.        While S ≠ NULL
28.            Do S <- Relax All unfound nodes for S.front(), connected using low power radios.
29.            If a High power radios can be connected b/w S.front() and Q.front() of which S.front()'s
                island is unfound
30.            Then Q <- S.front

```

```

31.          T <- S.front
32.          S.front becomes the new center for its island
33.          Update color, distance and precede as needed.
34.          Repeat Step 15 through 21 for the new center that was found in the previous step
              replacing SINK before
35.          color[island(S.front()) <- Gray
36.          S.pop().
37.          For all J in V[G]
38.              If color2[J] == Gray
39.                  Then color2[J] <- White
40.          Q.pop()

```

Let us discuss how the pseudo code works and find its complexity. We have three queues in our implementation. Steps 1-6 initialize all nodes in the network, and it executes a maximum of V times so the complexity is $O(V)$. There are 4 statements in the loop and V has to be multiplied by 4 but we are neglecting the constant 4, assuming large values of V . Steps 7-8 initializes all islands in the network. Let us assume we have one island and the complexity will be $O(1)$.

Steps 9-14 initialize the queues with, the sink and then change the properties of the sink. Then steps 15-21 form a loop in which all nodes in the sink's island are relaxed and added to the queue. Also their corresponding properties are updated. The same sequences of steps are later used to relax newly found centers in unfound islands. Now the complexity of the loop can be calculated by assuming that we have a single island, and its complexity is $O(V*V)$.

Steps 22-39 executes and tries to find the best possible solution for the problem defined. It has another new queue S which is initialized with all the centers of unfound islands and the center is marked as found. The first node in the queue S relaxes its neighboring nodes connecting on low power radios and adds it to the same queue. Also it tries to establish a connection between with the first node in Q , if it succeeds then the first node in S becomes its new center for its island else its popped out and the next node in the queue is taken and the above procedure is repeated until it explores all the island in the network. All the properties

of the nodes are updated as necessary. As we discussed before steps 15-21 is repeated here with the new center that was found in the previous step instead of sink as we are considering only one island for calculating the complexity so we eliminate the one outside the loop and considering the one inside the loop to get the lose bound . This loop repeats itself until queue S becomes empty. Finally before popping the first element in Q, color2 which is temporarily used is reset.

Now for the complexity of steps 22-39, we state the same assumption as before that we have only one island in the network. The loop 22-39 executes V times. Loop 27-36 executes itself 2V times. Step 34 is a loop which executes V times and steps 37-39 executes V times. So the complexity of steps 22-40 will be $O(V(V^2+V^3+ V))$. To get the total complexity we need to do a summation and it is in the order of $O(V^4)$.

We come with another Heuristic in the interface activation in the above algorithm where everything remains the same as above except that we are trying to activate the interfaces in the reverse manner. The algorithm includes finding of islands and finding its center along with it. We have also compared this heuristic along with the others.

Interface Activation (Heuristic II):

1. **For** Each U in V[G]
2. **Do**
3. Node[U].color <- White
4. Node[U].precede <- Negative
5. Node[U].distance <- infinite
6. Color2[U] <- white
7. **For** Each I in Island[G]
8. Island[I].color <- white
9. Get the total of unfound islands
10. T <- Sink
11. Node[Sink].color <- Gray
12. Node[Sink].distance <- Zero
13. Node[Sink].color <- Gray

```

14. Node[Sink].precede <- Negative
15. while T ≠ NULL
16.   For each W in Island[Sink]
17.     Do Relax of T.front()
18.     Update color, precede and distance as needed.
19.     T <- all relax nodes of T.front()
20.     T.pop()
21. while unfound islands > 0
22.   For all I in Island[G]
23.     If color[I] == white
24.       S <- I.center
25.       I.center <- Gray
26.   While S ≠ NULL
27.     Do S <- Relax All unfound nodes for S.front(), connected using low power radios.
28.   Q <- SINK
29. while Q ≠ NULL
30.   Do Q <- Relax All found nodes for Q.front(), based on the precedence that we already got.
31.   If a High power radios can be connected b/w S.front() and Q.front() of which S.front()'s
       island is unfound and Q.front()'s island is found.
32.     Then unfound--
33.       T <- S.front()
34.       S.front becomes the new center for its island
35.       Update color, distance and precede as needed.
36.       Relax T<- T.front with low power radios and explore the island with new center and
       update as needed and then T.pop().
37.       color[island(S.front())] <- Gray
38.       Q.pop().
39.       S.pop().

```

40. **For** all J in V[G]
41. **If** color2[J] == Gray
42. **Then** color2[J] <- White

Steps 21 – 42, runs until there is no unfound islands left in the graph. The difference in the pseudo-code can be seen in the main algorithm section of the code. Where in the first case the found islands explore the unfound islands, where as in the second case the unfound islands search for all found island to make a connection between them. Queue S is initializes every time with the list of all unfound island's center. Q is always initialized with the SINK. Q is relaxed based on the information of precedence of found islands that we will have. S is relaxed based on the shortest energy path with low power radios. Queue T is temporarily used to explore newly found island as soon as the new center has been determined. The temporary color2 used is refreshed back at the end just before the whole process starts up again to find the next unfound island.

The overall complexity of the algorithm has not been changed as only the order of processing has been changed and nothing else.

CHAPTER 6

RESULTS

Simulation Settings

The algorithm was implemented using C++. The nodes were generated randomly using Excel where the X axis and Y axis range from 0 – 2000. The test cases were done by varying the density of the nodes from 100-500 in the increments of 50. We did five trials for each density of the network and considered its average to plot the graph, to avoid too much variation in the trials. The set of random network topology chosen for test cases where made sure that they were distributed throughout the grids for better results.

Performance Metric

The metrics used to test were the number of high power edges turned on, average path length in the network and average hop count in the network. The reasons for choosing them are as follows

1. High power Radios turned on - If the overall number of high power radios that are turned on were reduced, then it reduces overall energy consumed by the network, which helps in minimizing the energy consumption in the network. So, it is good to have the number of high power radios turned on in the network to be minimum.
2. Average energy path - Energy path calculates the energy cost along the path. The energy consumed along the path from node to reach the sink has to be minimal; minimizing individual

energy path reduces the average energy path. We achieve our next goal only when the average energy along the path is minimal.

3. Average Hop count – Hop count is the count of the number of nodes along the path of a node to reach the sink. The hop count for each node has to be minimal; when it is minimal the average hop count is also minimal. This reduces the latency of the network which helps in transferring data to the sink as quick as possible.

Methods compared

We are comparing our performance metrics on the following methods

1. Minimum spanning Tree - Implementing the minimum spanning tree helps in reducing the overall energy consumed in the network, i.e. it reduces the number of high power radios turned on in the network. Prim's method was chosen to implement a minimum spanning tree and the three performances metric. We picked minimum spanning tree for the first metrics as it gives the best solution to reduce the number of high power edges turned on
2. Shortest path – Implementing the shortest path give the lowest average hop count possible, and tries to minimize the average energy cost path. Dijkstra's method was chosen to implement shortest path and evaluate the metrics. We picked shortest path for the best results of second the third metrics.
3. Proposed Heuristic I and II – We came up with a heuristic that gets a balance between these metrics which has been illustrated using a graph above. Also the pseudo-code of the algorithm has been given and explained.

Experiment 1

We compared the Total HP radios turned on in the network all the three methods which can be illustrated in the graph below

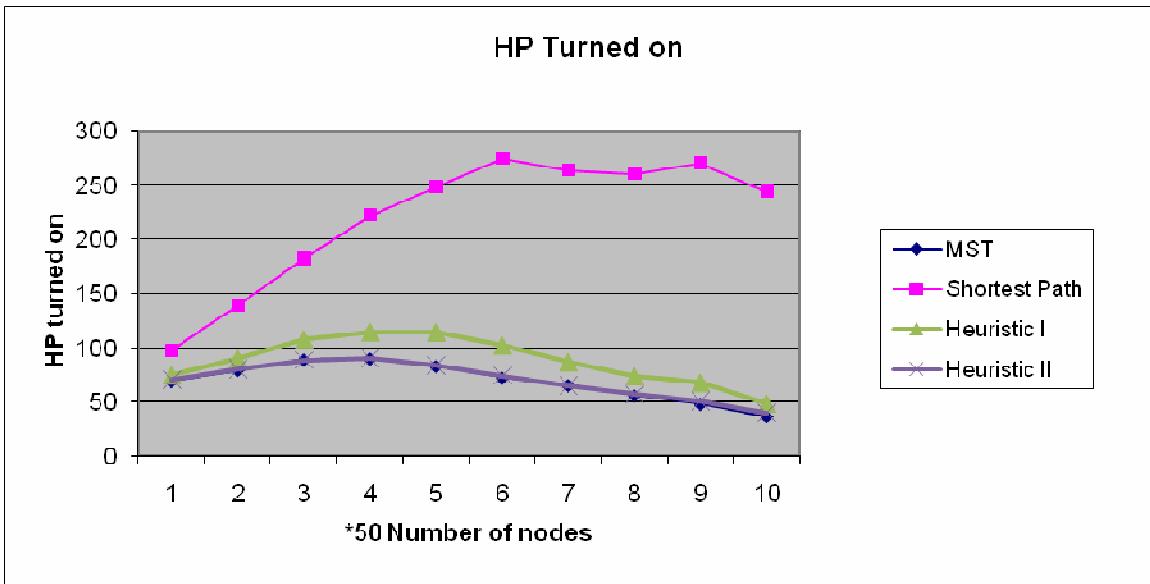


Fig 6 - HP radios turned on

From Fig 6 we can say that the number of High power radios turned on the network using shortest path is much higher than MST and our heuristics. Our heuristics tries to be as close to the MST as possible. MST gives the minimum HP turned on in the network. Now from the other two graph it can clearly be illustrated what we tried to do. MST gives the minimum HP turned on possible. Heuristic II gave better results then Heuristics I in this experiment.

Experiment 2

The second experiment was to find the Average energy path, the graph is shown in Fig 7. We implemented shortest path as the base for path length and it had the shortest path as a result minimum average energy path. The energy along the path for each node to reach the base station was found and average was taken to plot the graph. The average path length shown on the graph is based on the energy along the path. MST, which had the lowest number of HP radios turned on had a high average energy path. From the graph we can show that the average energy path of our heuristic I was almost closer to that of shortest path. Heuristic II did not have much control over the average Energy path and it was much fluctuating.

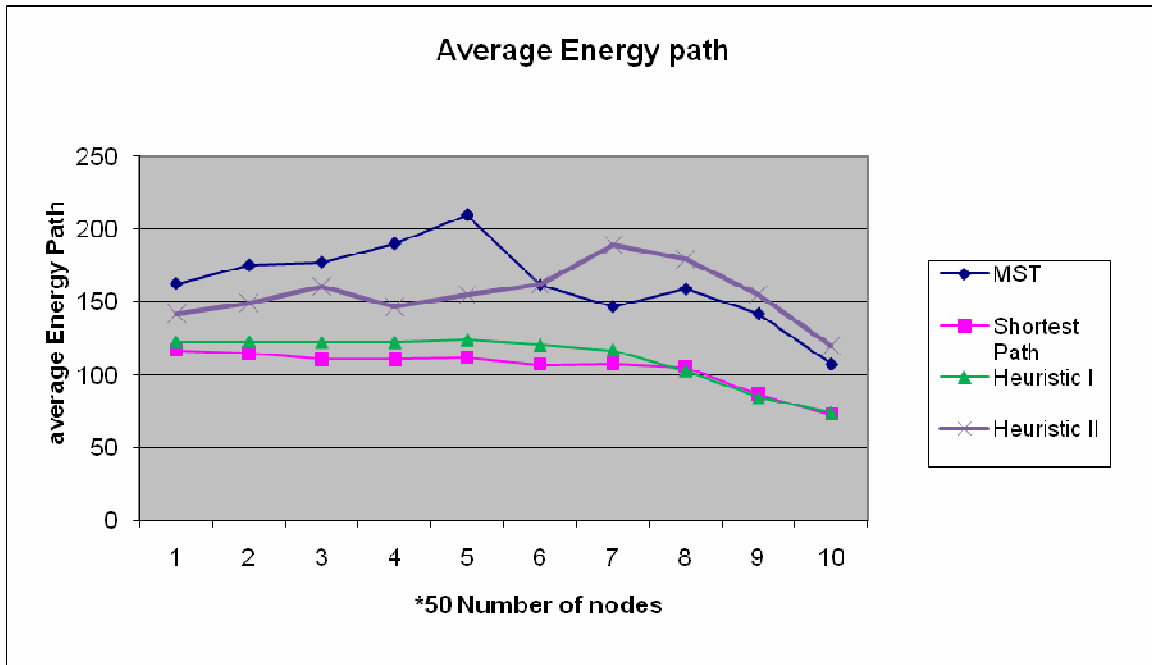


Fig 7 – Average energy path

Experiment 3

The third experiment finds the hop count to determine the latency in all the four methods. Lower the hop count lower the latency of the network. The number of hops each node take to reach the base station is the hop count for that node. The average was found for the network and graph was plotted.

From Fig 8 we can see that the shortest path has the minimum average hop count in the network and MST has the highest. Our heuristics give a result as close to shortest path as possible. As the density of the nodes was increased the hop count increases. Both Heuristic did the best to achieve the minimum average hop count as possible for the given network.

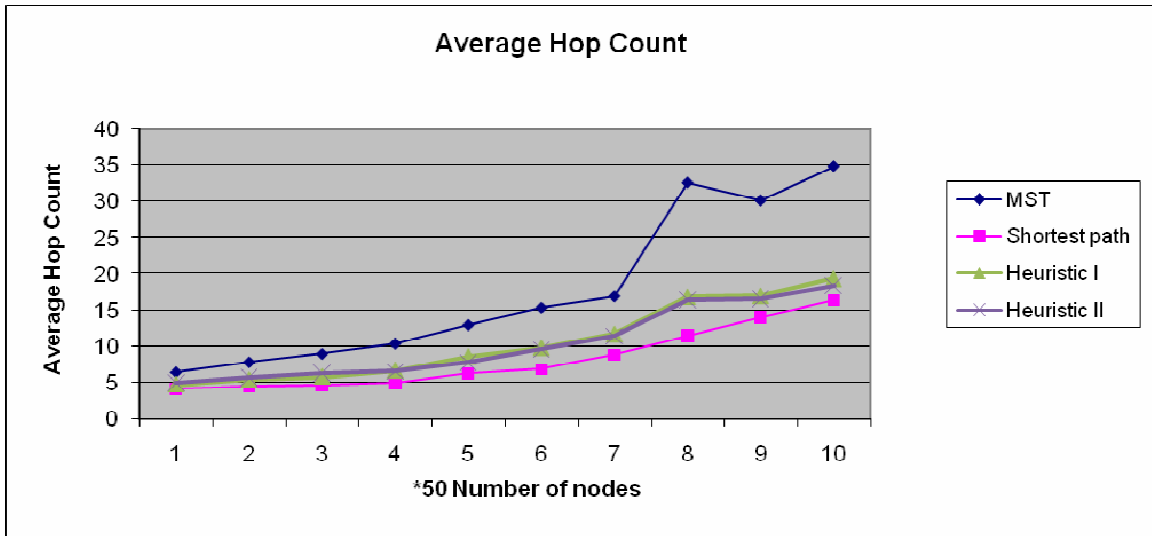


Fig 8 – Average Hop Count

From All the three graphs we can say that MST gives the best result for minimizing the number of HP radios turned on however its Average hop count and average energy path is high. Shortest path gives the best result for minimizing average hop count and average energy path however the number of Hp radios turned is much high. Our heuristics tries gives a balance between these metric which has the number of HP radios turned on the network close to that of MST and at the same time has the average energy path and average hop count close to the shortest path. Our heuristics meets the goals described earlier and has a complexity in the order of $O(V^4)$. However Heuristic I is better than Heuristic II because it has a better balance in the graphs. As we can see in the second graph where the Heuristic II did not have control over the energy path.

SUMMARY

The problem about interface activation on wireless radio networks was discussed and the importance was illustrated. The problem was defined and explained and a heuristic was proposed and implemented. Also the heuristic was compared with the existing basic algorithms to differentiate the efficiency and its complexity. The graphs were shown on how effective our heuristic can be depending on the density of the network. Our overall goal was to minimize the energy consumption and minimizing the path length between nodes to reach the sink. That has been achieved from the results. We have increased the number of turned on radios slightly compared to the minimum spanning tree and achieved the average path length almost equal to shortest path. The pseudo code of the algorithm implemented has been given and explained on how it works and its complexity calculated to be in the order of $O(V^4)$. However we can see that the graph narrows down as the density of the network increases that is because as the density increases at one point there will be just one island in the network and shortest path will be the best possible solution for it. The possibility of minimum spanning tree to reduce its average path length increases.

The implementation was considered for two radios in a node. This can be extended to K radios by starting to map the lowest power radios and they mapping in ascending power of radios in the nodes. However there will be different range of islands in the network depending on the power considered and the complexity grows with respect to the number of radios.

This problem can be translated into a kind of group Steiner problem, and it's a NP-hard problem [3]. Solving this problem in a distributed way should give a better result however reducing its complexity might be an issue to be considered. This problem is different from the existing group Steiner problems by not allowing to add any new nodes in the network and the network have to be connected with the given topology which is random. Also working on the K radios can be implemented and its complexity found as a future work to this problem.

REFERENCES

- [1] I.F. Akyildiz, W.su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Volume 38, Issue 4, Pages 393-422, 2002 Published by Elsevier Science B.V.*
- [2] Mohamed Younis and Kemal Akkaya, "Strategies and Techniques for Node Placement in Wireless Sensor Networks : A Survey," *Volume 6, Issue 4, Pages 621-655 May, 2007 Published by Elsevier Science B.V.*
- [3] Mo Li and Yunhao Liu, "Underground Structure Monitoring with Wireless Sensor Networks," *in IPSN'07, April 25-27, 2007.*
- [4] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler and John Anderson, "Wireless Sensor Networks for Habitat Monitoring," *in WSNA, September 28, 2002.*
- [5] Dimitrios LyMBERopoulos, Nissanka B. Priyantha, Michel Goraczko and Feng Zhao, "Towards Energy Efficient Design of Multi-radio Platforms for Wireless Sensor Networks," *in (ISPN 2008) - Volume 00 Pages 257-268, 2008.*
- [6] Carlos E. Ferreira and Fernando M. de Oliveira Filho, "Some Formulations for the Group Steiner Tree Problem," *in Discrete Mathematics 18 (2004) 127-132.*
- [7] Richard Draves, Jitendra Padhye and Brian Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," *in MobiCom, Sept. 26 - Oct. 1, 2004.*
- [8] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. B. Srivastava, "Topology Management for Sensor Networks: Exploiting Latency and Density," *in MobiHoc, 2002.*
- [9] Thanos Stathopoulos, Martin Lukac, Dustin McIntire, John Heidemann, Deborah Estrin and William J. Kaiser, "End-to-end Routing for Dual- Radio Sensor Networks," *in Proceedings of INFOCOM, 2007.*

[10] J. Sorber, N. Banerjee, M. D. Corner and S. Rollins, "Turducken: Hierarchical Power Management for Mobile Devices," in *MobiSys*, 2005.

[11] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *Mobicom*, 2002.

[12] T. Pering, V. Raghunathan and R. Want, "Exploiting Radio Hierarchies for Power-efficient Wireless Discovery and Connection Setup," in *18th International Conference on VLSI Design*, 2005.

VITA

Vijayaraja Sahadevan

Master of Science

Thesis: INTERFACE ACTIVATION IN MULTI – RADIO WIRELESS NETWORKS

Major Field: Computer Science

Biographical:

Personal Data:

Born on July 17, 1984 in India

Education:

Completed Bachelor of Engineering in Computer Science and Engineering at Sriram Engineering College, Anna University, Madras, India in May, 2005.

Completed the requirements for the Master of Science or Arts in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2008.

Experience:

Part time operator, IT Operations OSU, Stillwater OK. 2006-2008

Graduate Teaching Assistant, Computer Science Department, OSU, Stillwater OK. 2007 – 2008.

Professional Memberships:

Student member of ACM.

Name: Vijayaraja Sahadevan

Date of Degree: July, 2008

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: INTERFACE ACTIVATION IN MULTI – RADIO WIRELESS NETWORKS

Pages in Study: 24

Candidate for the Degree of Master of Science

Major Field: Computer Science

Wireless sensor network is used in a wide range of applications, and the technological advancements have allowed of having more than one radio in a sensor node. Wireless radio networks have wide range of functionality which can be used in places to get data and send to a monitoring base station constantly in certain intervals of time or when needed where manned work might seem hard. There are many issues to be considered in a wireless sensor network of which power consumption is one of the major issues. It is important for a sensor node to stay active for longer time in order to have the network connected. The nodes cannot be recharged every time, where energy has to be used more efficiently in every sensor nodes. Our goal is to minimize the total energy consumption and energy consumption along the path from the node to the base in a wireless multi-radio network and also reduce the latency for each sensor node to send data to the Base station, which means the data, can be sent to the base as quickly as possible without any delay. The problem has been defined for N radios in each sensor node and a sensor network with two radios has been considered for describing the problem and differentiating it from the existing algorithms. A plausible solution to the problem has also been proposed and implemented and their results compared with existing algorithms.

ADVISER'S APPROVAL: Dr. Venkatesh Sarangan
