Practical Approach on Communication Wireless Sensor Network Design for Engineering Education

Hamid Majlesein, Sujeet Bhatte, Zhengmao Ye
Department Electrical Engineering
Southern University and A&M College
Baton Rouge, LA 70813, USA

Abstract

Wireless sensor networks are one of the foremost areas of research in computer networking. The increased interest in this new breed of networks has risen due to their immense potential in various areas of application ranging from tracking birds to sensing poisonous and potentially lethal gases in subways and bus stations, especially after the increased risks posed by such terrorist attacks. Due to the nature of most applications these networks require to consist of very small, energy efficient elements which are able to make the most of their resources.

Many methods have been developed in order to reduce the amount of energy consumed in wireless sensor networks. This thesis examines one such method defined to help conserve energy while aggregating data. The method is examined with a standard wireless network standard, Pico radio and the amount of energy consumed in it is analyzed.

The nature of pseudo stationary networks is discussed and then the role of a method like Data Funneling in conjunction with such networks is analyzed. An algorithm called Zone Efficient Localized Data Aggregation Algorithm is defined which undertakes the facts associated with pseudo stationary networks into consideration and uses the advantages of Data funneling in order to maximize energy conservation. It is the aim of this thesis to show how energy can be conserved by minimizing the number of nodes involved in communication in a pseudo stationary network.

Introduction

In the case of zoning algorithm used in the Data funneling algorithm, the zoning occurs during the discovery phase every time the network is queried for information. The amount of packets needed to zone the entire network is not extremely efficient due to the following factors:

1. During the discovery phase, the entire network gets flooded with discovery packets. The amount of network congestion caused by such heavy number of data packets cripples the network for the amount of time this entire process takes.

2. The discovery packet at each phase needs to be examined by the individual node which amounts to computational energy which is utilized by that node irrespective of it's involvement in the point to point communication between the requesting node and the queried zone.

3. In the case of applications where the nodes are pseudo stationery and the area and location of the canvas is also almost non-transitioning, the zonal coordinates seldom change and therefore the need to zone or discover the network during every query is not required.

4. Due to the fact that the discovery process occurs every time a network region is queried, there is a residual latency between the request and the information gathering which is a result of the discovery process. This latency if possible could be done away with, thus increasing the time efficiency in gathering information in such networks.

BASICS OF CUBOID

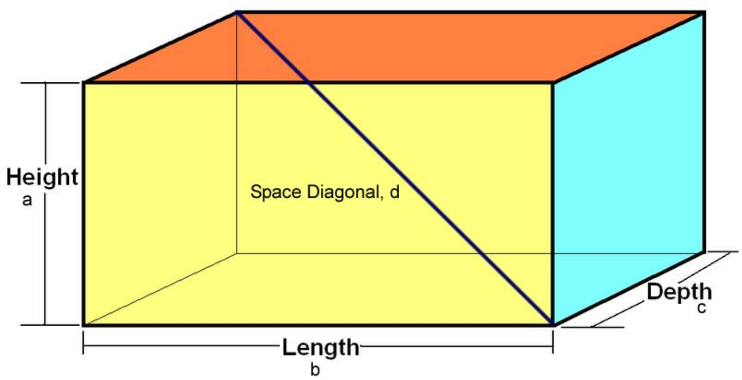

Figure 1. A Cuboid with Dimensions (a, b and c)

In geometry, a cuboid is a solid figure bounded by six rectangular faces: a rectangular box. All angles are right angles, and opposite faces of a cuboid are equal. It is also referred to as a rectangular parallelepiped. If the dimensions of a cuboid are a, b and c, then its volume is *abc* and its surface area is $2ab + 2bc + 2ac$. The length of the space diagonal of a cuboid = $d = \sqrt{a^2 + b^2 + c^2}$.

In case of a perfect cuboid, where all the sides of the cuboid have equal value, the space diagonal will be = $\sqrt{(3*l)}$ where *l* is the length of the side. Let us consider a cuboid with space diagonal vertices A and G. The coordinates of the two vertices are (X_A, Y_A, Z_A) and (X_B, Y_B, Z_B) respectively. Therefore, in case of a perfect cuboid, if we know any two space-diagonally opposite coordinates, then we can identify the entire volume of the cuboid.

ZONING EFFICIENT LOCALIZED DATA AGGREGATION

Zoning Efficient Localized Data Aggregation with Data Funneling is defined as an approach to resolve the inadequacies of the default zoning algorithm [1] with respect to power consumption for pseudo stationary applications. It utilizes the fact that the sensors in the network are pseudo stationary in nature and that the entire network structure does not exceed a predefined area. This is true in applications like Environmental Sensing, Health monitoring, Security Monitoring where the sensors though mobile may not change location very often and if they do then the difference in location between previous and current location is very small.

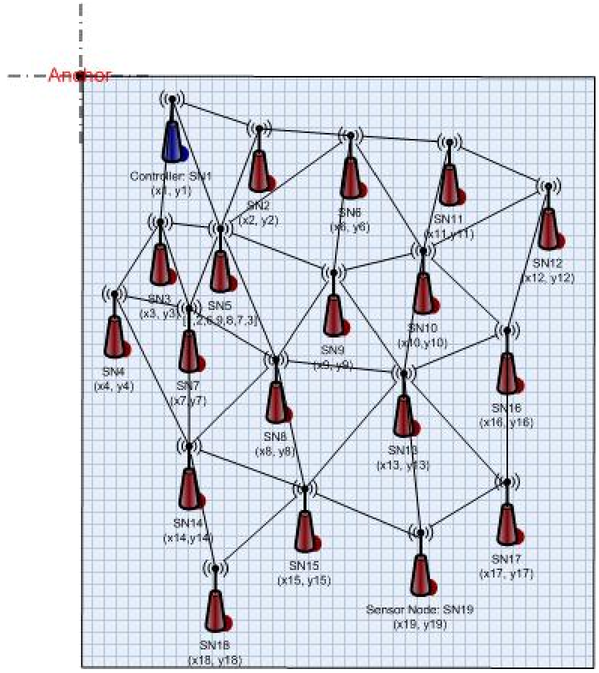

Figure 2. Directional Zoning Anchor Representation

As discussed by Petrovic[1], the location of the sensors, the temporal sequence and sensor information for the readings are undertaken are irrelevant since all the controller is looking for is the readings from a designate zone for a predetermined period identified at the moment the controller queries the network. Therefore, it is not necessary to have a method like the one used in Data funneling to discover and zone the entire network.

This is not needed simply because of the fact that the network structure will in most probability stay the same between two requests for information by controllers in the network. It can be safely said that, though the nodes are wireless and ad hoc in nature, the difference in location Δd between current and previous locations is so small that it can be neglected. Besides the fact these nodes are sometimes limited in motion within a region in the area of operation. For example, if a sensor network is employed for ambient temperature control in an office building, then the sensors will be limited by the physical walls for the rooms inside the building. In such applications, wireless sensors are extensively used off recent since they are much cheaper and easier to deploy.

In the case under consideration, since the network area of operation is fixed, there is no need of discovery every time a query is executed other than at wake up due to the fact that the network structure doesn't change very often. ZELDA takes a different approach compared to Data Funneling for zoning and communication between nodes within a given designated network.

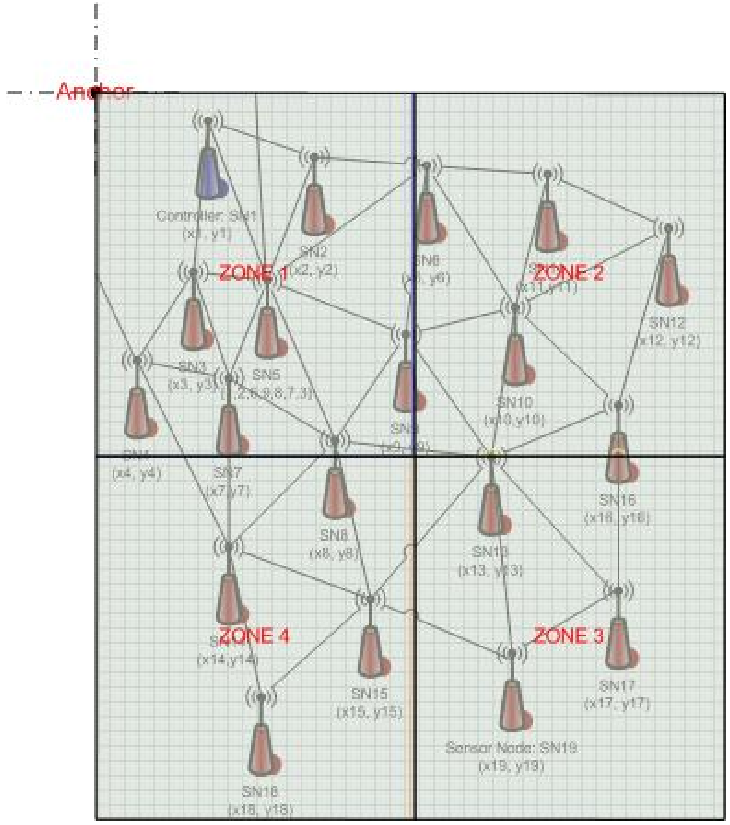

Figure 3. Zoned Network using Zoning Efficient Localized Data Aggregation

ZONING IN ZELDA

ZELDA is proactive algorithm. The ZELDA algorithm runs during the discovery phase of the network and zones the network into perfect cuboids using the anchor nodes added to the network. The process is defined in detail as follows:

1. Discovery Phase: The discovery phase of ZELDA is similar to Data funneling. On wake up, each node finds its nearest neighbors and aggregates a list of all the nearest neighbors. However, this discovery is limited to nodes of a particular network only.
2. Zoning: A fixed anchor is designated within the area of the network. Using this point, the entire area of the network is divided into perfect cuboids. These cuboids identify the zones in the network. Each sensor maintains the coordinate pair of the space diagonal opposite vertices in order to identify its zone. This is referred to as the zone ID.
3. Since the sensors are pseudo stationary in nature, they do not change zones. In case a sensor changes a zone it is made aware of the coordinate pairs for the zone it belongs to by its neighboring nodes.
4. When a controller needs to request information from a zone in the sensor network, it directionally floods towards the area of interest using the Cartesian coordinates of the cuboid in which it exists.
5. After the interest packet reaches the interest zone, it follows the same procedure that is followed in Data funneling.

Instead of having to divide the entire network into zones and have the overhead of maintaining and updating the relevant information related to it, we simply maintain the basic structure of the area of operation at the controller node. This information basically includes dimensions of the said area of operation. In the kind of applications [1] that employ the Data Funneling algorithm this will not change in most cases. When we need to gather data related to a part of the network, we simply provide the relative coordinates of a cuboid which will cover that region.

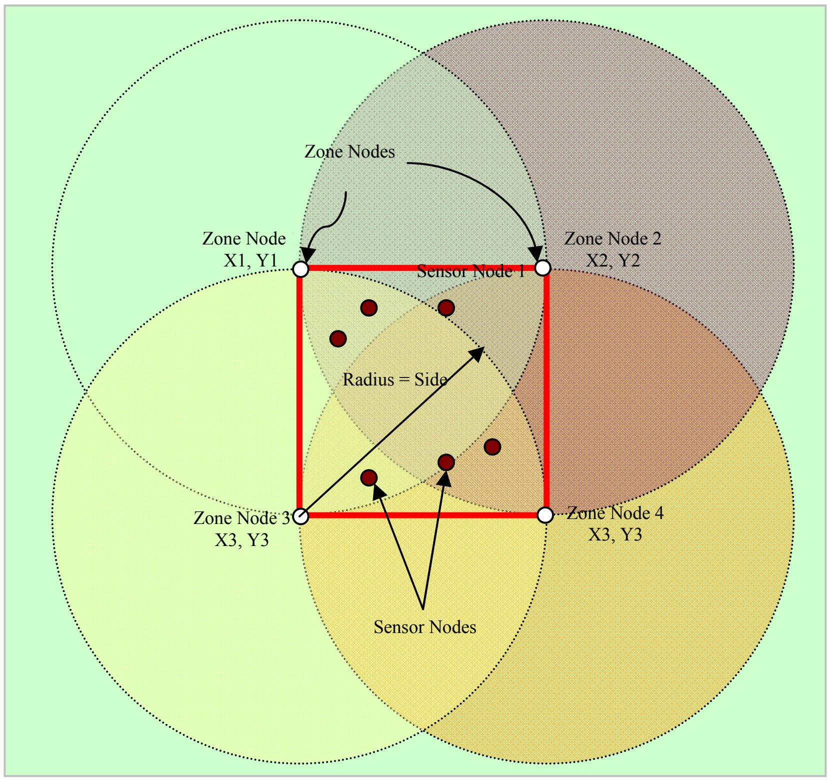

Figure 4. Zoning Using Radio Coverage Radii in ZELDA

Figure 4 shows an example in which the cuboid has all sides equal and therefore is a perfect cuboid such that the length of the side is equal to the maximum radius of the zonal nodes. Therefore, the four zonal nodes create an area of common coverage. The sensor nodes falling in this zone can be represented by a pair of Cartesian coordinates of any two nodes. Each zone node has a unique set of Cartesian coordinates. Sensor node 1 shown in Figure 4 above falls in the area of coverage of zone nodes 1, 2 and 3. However, since zone nodes 1 and 3 are diagonally opposite to each other, the zone ID representing the zone for sensor node 1 will the be the pair of Cartesian coordinates for 1 and 3. The zone ID for sensor 1 will be [(X1, Y1), (X3, Y3)].

In order to find the temperature at the Northeast corner of the building for a room the algorithm is given the interest packet with coordinates of the two diagonally opposite vertices of the cuboid. The network can then be directionally flooded in any given direction using directional gain for antennae in case of a wireless sensor network. This may be difficult in case of a wired network or complicated due to the nature of the same. Every node checks its coordinates with those supplied by the interest packet and if it falls within those coordinates, it uses a flag to indicate the same. At every node the hop count to the controller is updated[1].

This process eliminates the need to discover the entire network every time a request is sent. Therefore, the amount of latency between request and information gathering is reduced. It also reduces the energy cost associated with zoning in the case of Data funneling which is substantial if the controllers require information on the fly and the intervals exceed the amount of time the nodes stay awake. Besides in the case of pseudo stationary wireless networks in which the area of operation is fixed, there is no need of discovering the network topology every time a request is sent. An alternative to this could be to update the zone list with node numbers within the zones and maintaining this list with every node. However, since it is irrelevant to the controller what node resides in which zone this may not be done. In order to make sure that the nodes stay up to date with respect to the zone IDs they belong to, a periodical update may be undertaken. However, this update is optional and is dependent if the network changes at all.

ZONE EFFICIENT LOCALIZATION AND DATA AGGREGATION

We define the Zone Efficient Localization and Data Aggregation Algorithm (ZELDA) in an attempt to optimize the resource management in a wireless sensor network which is pseudo stationary in nature and has a fixed area of operation. The primary areas of energy utilization have been identified in the data funneling algorithm to be due to the main fact that the algorithm requires the entire network to be discovered every time a request is sent from a controller to a sensor. The network undergoes all the initial phases of initialization as if it were just formed. This process is very advantageous when used with networks with high node volatility and continuously changing configuration. The algorithm also uses a Max Min D clustering type algorithm to create the zones which in itself creates a power utilization scenario since this process is undergone every single time the network operates. Since the network is reactive in nature it is easy to presume that this happens almost every time since the amount of wake time for a network will be very small as compared to the frequency at which the data is being requested. Especially in applications like temperature sensing, this process can be very ad hoc in nature.

In the case of ZELDA, we divide the entire physical volume of operation into cuboids. This is quite possible and feasible since the volume if known and physically unchanged during the entire operation of the network. For example in the case of an ambient temperature sensing application for a building the volume of operation is limited by the structure of the building. This is the volume of operation and it can be divided into cuboids as explained earlier.

ZELDA recommends adding zone nodes to the as shown in Figure 5 show below. Listed below are the steps involved in Zoning Efficient Localized Data Aggregation Algorithm

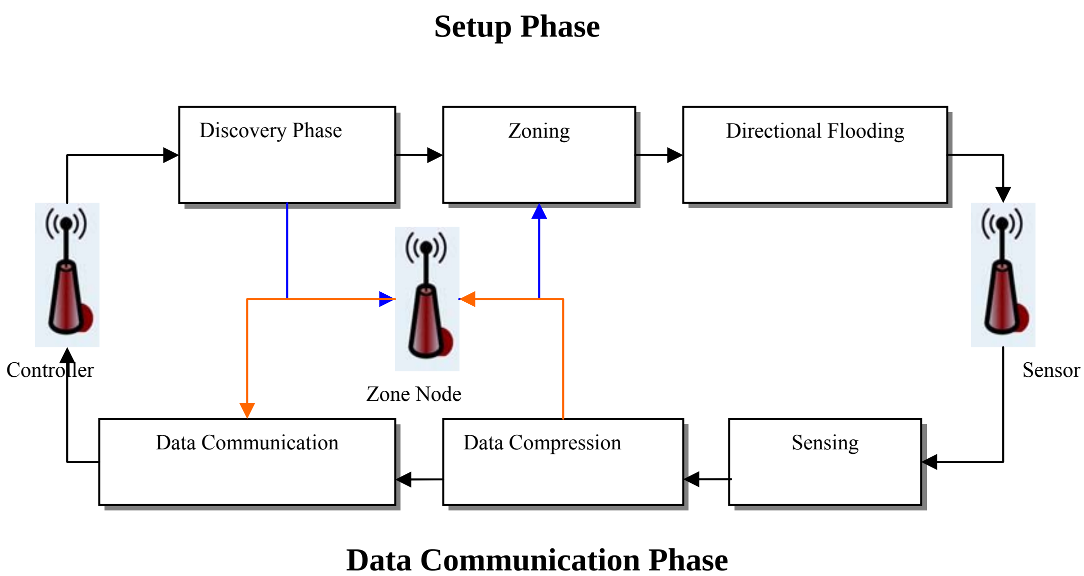

Figure 5. ZELDA Data Flow Diagram

DISCOVERY

On wake up, each node advertises itself to its nearest neighbors. It aggregates a list of nearest neighbors. The zonal nodes advertise within their zone of operation. They identify themselves by their Cartesian coordinates X, Y and Z. this information is given to the nearest neighboring sensor nodes. The discovery process does not differ much as compared to that of the data funneling algorithm, except for that fact that this occurs only once in the case of ZELDA. This process may be repeated in case a node changes location. However, since in our application the nodes are pseudo stationary in nature, this process will not be repeated very often.
Controller nodes are setup using the same logic used in data funneling and may be established using a standard cluster head algorithm as described in chapter 3.

ZONING

The sensor nodes will receive such information from two different zonal nodes at the same time. These two zonal nodes are the corresponding space diagonal opposite coordinates in the cuboid zone to which the sensor nodes belong. The sensor network registers this zone ID as the pair of Cartesian coordinates of the respective zonal nodes. Therefore, the zonal nodes create a virtual cuboid within which the nodes will lie. This is represented in Figure 6 below.

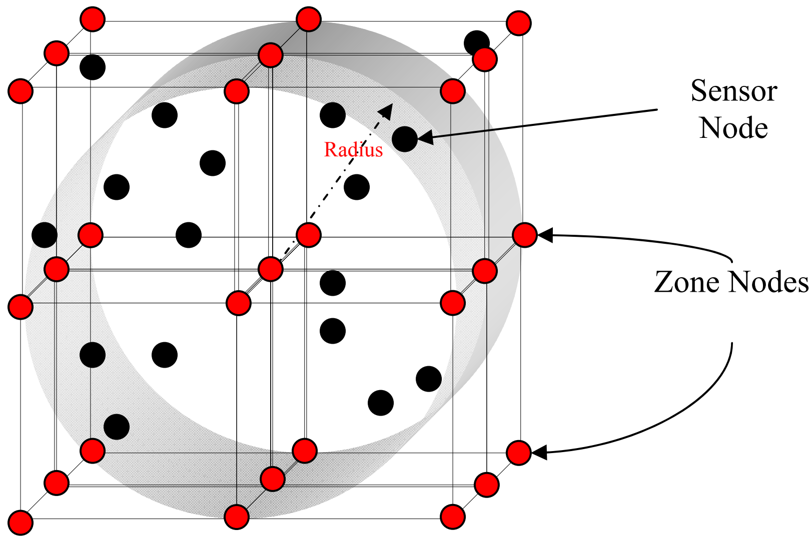

Figure 6. Zone Nodes, Radius of Operation and Sensor Nodes

Every zone node pair, floods its region such that all the sensor nodes falling within the respective zone know what zone they belong too.

The zone nodes in then communicate with each other identifying themselves and their position to other zone nodes directly.

DIRECTIONAL FLOODING

Every zonal node is aware of the zonal nodes in direct neighborhood. When a controller wants to communicate with a particular zone, it floods its own zone with the interest packet. This packet propagates through the controllers own zone until it reaches the zonal node. The zonal nodes intercept this packet and examine it for the area of interest which is indicated by the zone ID. They check if they fall within the area of interest in any of the adjacent cuboids that they are a part of. If they do not represent the zone ID being requested, then they add a field to the packet called the Controller Zone ID. This represents the controller's home zone ID and is the pair of Cartesian coordinates which represent it. This pair can be any one of the four available. All four Cartesian coordinates maybe sent in order to allow redundancy.

Instead of propagating the packet through the entire network, the zonal nodes only send the packet ahead to the nearest neighboring zonal nodes in the same direction. Therefore, the directional flooding utilizes only the zonal node infrastructure. Since the amount of zonal nodes is limited and the only purpose of the same is to administer the network, this process is much faster as compared to forwarding the packets through the network. The controller is not interested in individual sensors but a zone as a whole and therefore it is not required that the propagation occur through the sensor nodes. Every zonal node has only seven direct neighbors in the same zone and twenty four direct neighbors. All nodes are fixed in location. Directional flooding is more effective in this case since we know the direction in which a node is on the Cartesian coordinate system. So as long as we have the Zone ID we can forward the interest packet to the zone using this method. Every time a packet passes through a zonal node, the node updates the hop count and cost field in the interest packet.

Once the packet reaches the interest zone, the interest packet is stripped by the zonal node that receives it. The sensor nodes closest to it are designated as the border nodes and are responsible for aggregating and sending the data back to the zonal node. The zonal node sends a zonal interest packet with the hop count and cost fields reset. These fields now represent the hop count and cost to the zonal nodes. The zonal nodes are responsible for sending back the information to the controller through the zonal node network. There are four zonal nodes in one direction and therefore, even if the request comes through one of these four nodes, any other node from these can act as the aggregation and retransmission point based on a schedule that they may define. This allows the zonal nodes to conserve energy and make sure that not a single node is burdened by the responsibility to retransmit. This allows an equal decay of the system with respect to battery life and prolongs the life of the zonal network. Though the zonal network is an infrastructure network it may not be possible to have direct power supply and it is desirable to save as much power as we can. There are several advantages to this approach as compared to the one used in Data funneling. These are discussed in detail later.

SENSING

The sensors will sense the data for the predefined period of time. Then they package this data and in a packet like the one explained in Appendix A for temperature sensing. This packet is then transmitted from the sensor to its neighbor until it reaches the zone node.

DATA COMPRESSION

ZELDA can be used in conjunction with a data compression technique which will exploit the fact that the sequence of the readings and the sensor information from which they are received is irrelevant to the requesting controller and strip, concatenate and update the super packets with information in the same manner that the data compression employed in Data funneling does. For sake of simplicity, we consider the same technique assumed in the explanation for data funneling in chapter 3. This will enable us to better compare the two techniques for energy efficiency.

Future work is recommended in investigating the obvious gain in power consumption achieved while using various compression techniques with ZELDA.

DATA COMMUNICATION

Zone nodes are stationary in nature and are located at the vertices of the cuboids in which the entire network is divided. They are placed at the X, Y, Z coordinate locations of the same and do not change location for the entire period of the network existence. The zone nodes have the coordinates hard coded into their firmware and they use this value to represent themselves to other nodes.

On network wakeup, just like the sensor nodes, the zone nodes transmit their information to all nodes which fall within their network range. Every zone is identified by at least two zonal nodes and it is the region in which they both have common coverage. That means their 3 dimensional coverage area intersects in that given physical zone. The sensor nodes within that zone receive the Cartesian coordinates of each and will identify themselves using this pair as the Zone ID.

Instead of having to divide the entire network into zones and have the overhead of maintaining and updating the relevant information related to it, we simply maintain the basic structure of the area of operation at the controller node. This information basically includes dimensions of the said area of operation. In the kind of applications1 that employ the Data Funneling algorithm this will not change in most cases. When we need to gather data related to a part of the network, we simply provide the relative coordinates of a cuboid which will cover that region.

If we compare the energy equation from chapter 3, we have

$$\eta = \frac{EZELDA}{EDF} = \frac{C1 + ES + ECS + E\,L + R*(EDirF + EDComm)}{C2 + R*(ES + ECS + EZ + EDirF + EDComm)}$$

This equation gives the relationship between energy consumed in Data funneling and Data funneling when using ZELDA. An analytical investigation related to energy consumption is undertaken in the next chapter where both algorithms are compared using standard assumptions and metrics.

ROUTING AT THE ZONE NODES

The zone nodes are used for localization and packet transfer in this algorithm. There is no need for switching or broadcasting since the zone nodes for a perfect mesh. In this scenario, we will consider the communication between node A and node B as shown.

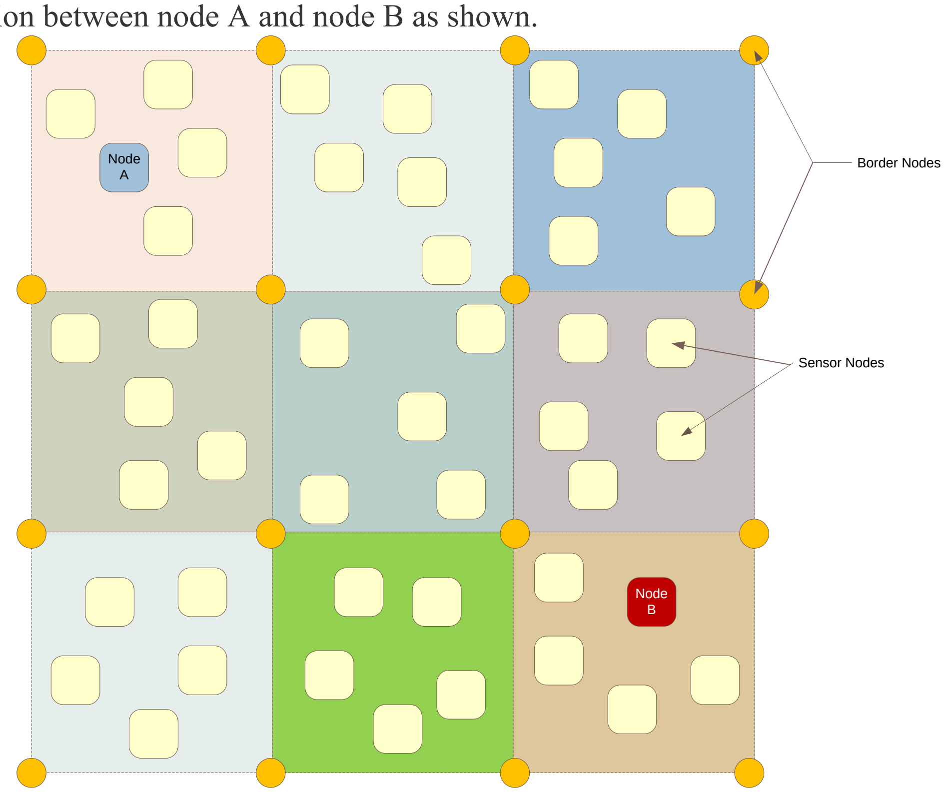

Figure 7. An Example Sensor Bed for ZELDA Algorithm

As you can see in the Figure 7, the sensors are indicated by yellow squares. These sensor nodes are placed in an area divided into square zones shown in different colors to distinguish between each other. The border nodes are shown at the four corners of each zone as per the scheme for ZELDA shown by orange circles.

It is assumed that there are a maximum of five nodes at any given time in any zone. These random nodes are considered to be stationary. This model will be used to analyze the amount of energy consumed during the communication between node A and node B using ZELDA algorithm. In order to make the comparison easier, let us assume that the network is used for the ambient temperature sensing application.

INTER-REGIONAL FLOODING

When Node A needs to communicate with any other node, it first needs to advertise its location and intentions.

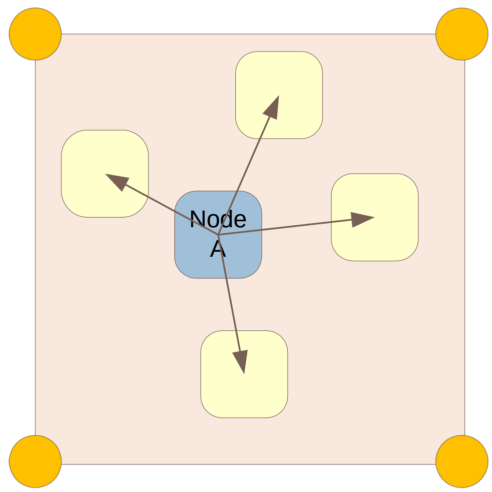

Figure 8. Inter Regional Flooding

THE RIGHT HAND RULE

Routing in the border nodes can be a challenge when communication between more than one zone is required. Primarily because of the fact that the communication has to occur concurrently while other nodes are trying to communicate with each other. In order to create order, the right hand rule can be utilized to identify the basic mechanism in which any border node at a given time will choose to transfer a packet to its nearest neighbor.

The right hand rule simply states that the border node will transmit the packet to the first available neighbor to its right in the XY plane with Z as reference. In the case that the node is not available or busy then the node will try to transmit to the node on the right in the YZ plane with X axis as the reference. So on and so forth. This simple method to resolve the next node eliminates the need to have an advanced routing algorithm in place while ensuring that the packet will be transmitted in a directionally sound manner.

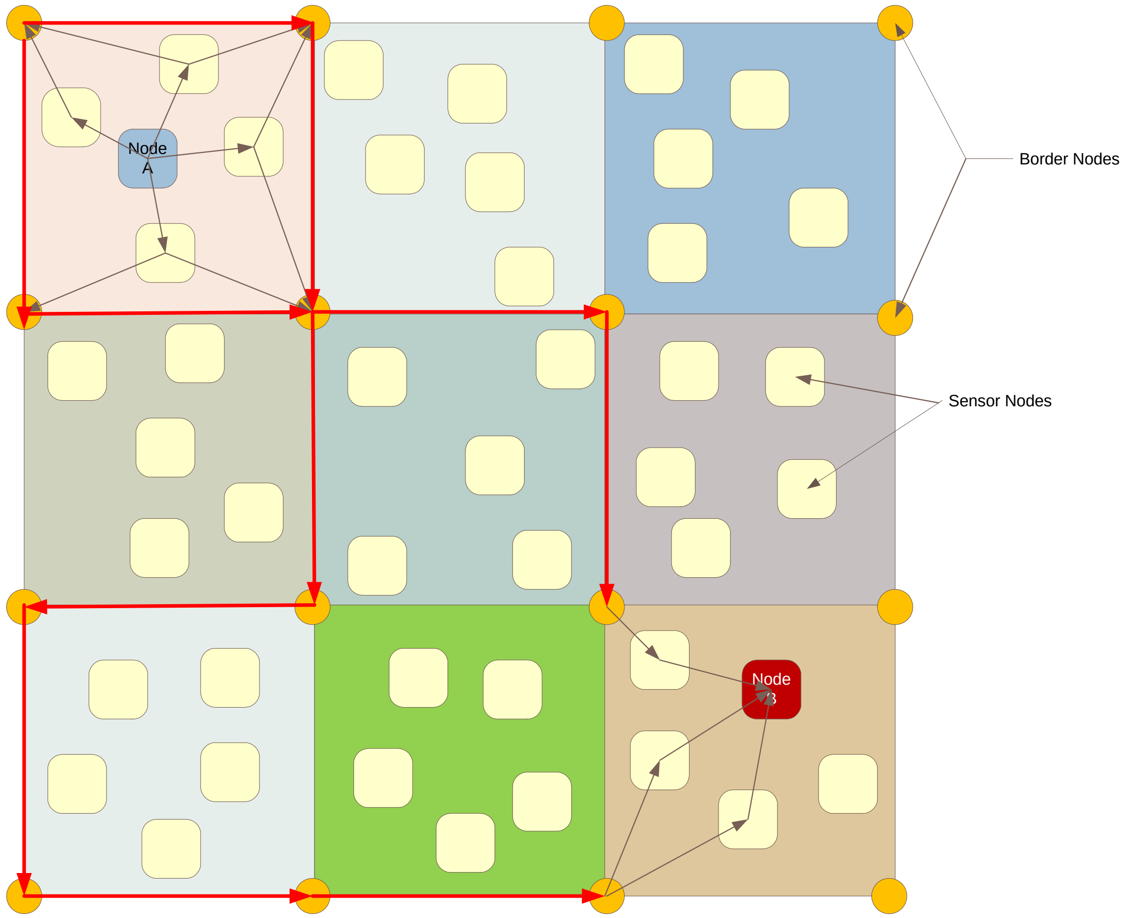

Figure 9. Directional Flooding

Above and Figure 6 above shows the setup phase for communication between node A and node B using ZELDA.

SETUP PHASE

The setup phase of the Data funneling algorithm and the ZELDA algorithm were described in detail earlier. As per the definition of each algorithm, a process is followed in the different phases of the algorithms. Here the setup phase will be examined in order to compare both the algorithms and deduce how energy consumption is achieved in ZELDA when compared to Data funneling algorithm. Here the assumption is that both algorithms employ the similar data compression techniques and at any given time, the packet size remains same. This holds true since fundamentally the packets used in both algorithms are the same. Therefore, the number of nodes used in the setup phase for each network is as plotted below.

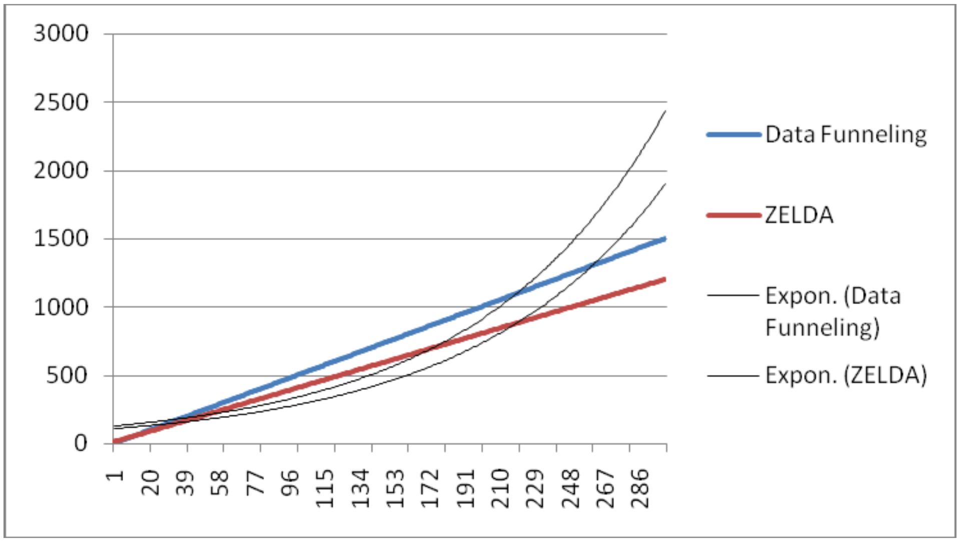

Figure 10. Nodes When Each Zone Has n =5

The above graph shows the plot for ZELDA and data funneling when the number of nodes in zone is equal to 5. For the ZELDA algorithm, it is assumed that this excludes the number of border nodes. It can be observed that the number of nodes used in the setup phase for Data funneling is considerably higher and the difference in the values between Data Funneling and ZELDA increases as the size of the network increases.

The exponential plot depicts the energy that will be consumed by the nodes for each algorithm. This graph clearly shows that there is considerable energy savings in ZELDA as compared to Data funneling algorithm in the setup phase.

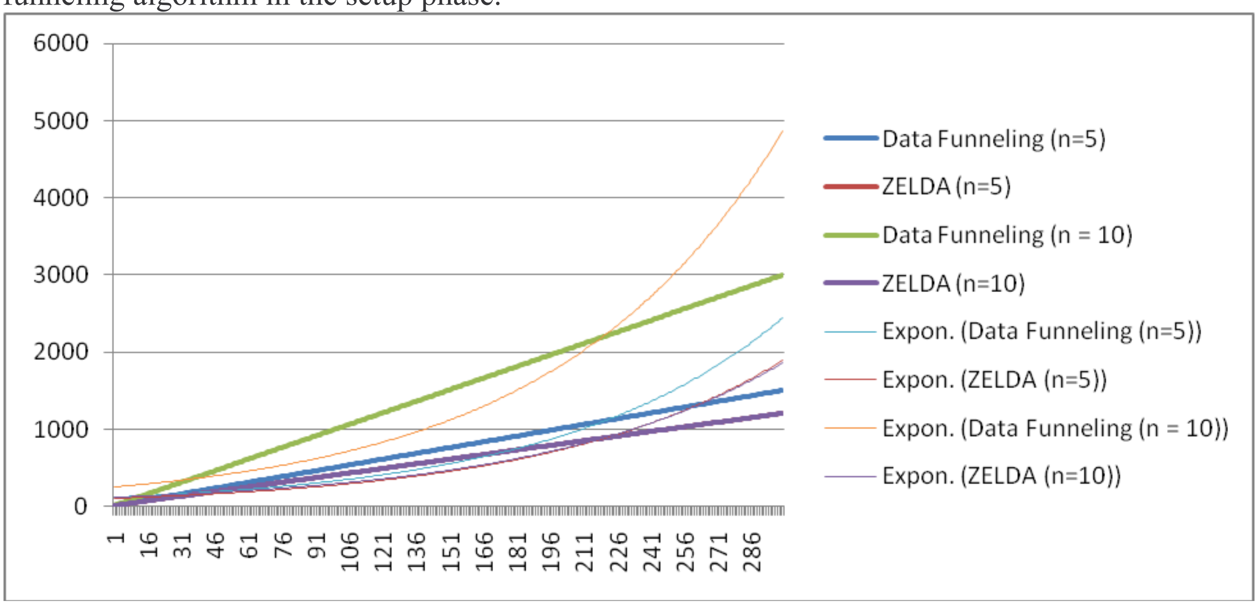

Figure 11. For Values of n = 5 and n = 10

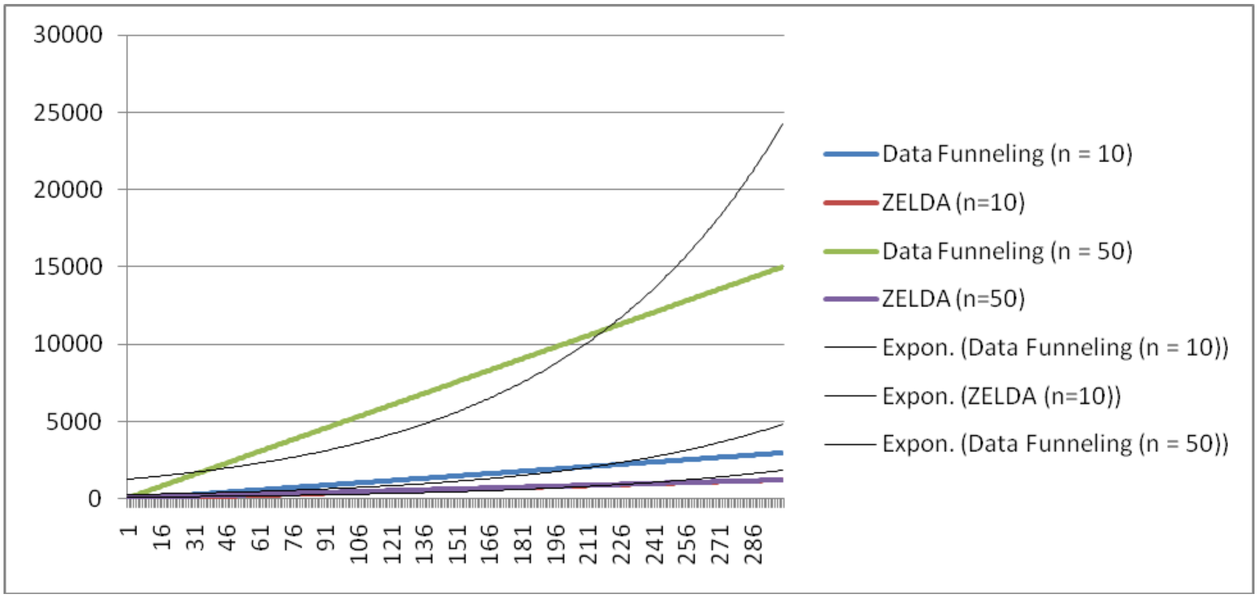

Figure 12. For Values of n=10 and n=50

Figures 11 and 12 show the comparison of values for Data funneling algorithm and ZELDA when the values of n are 5 and 10 and 10 and 50, respectively.

Summary and Conclusions

When a packet is transmitted in the network from one node to another, energy is expended at each node in order to process this packet and forward it to the next node. As a fundamental rule, if we are able to minimize the number of nodes involved during communication between two nodes in a network, then the overall energy consumed in this process will be reduced.

Here it can be observed that the comparative number of nodes for ZELDA almost remains the same even if the density of the network increases considerably. However, in the case of Data funneling algorithm, as the density of the network increases, the number of nodes utilized for setup increases steadily and at a higher rate. This directly translates into increased energy costs for the overall network. However, in case of ZELDA, the energy cost for setup stays the same no matter how large of small the network node density is. Therefore, the amount of energy dissipated during setup and discovery stays the same. Based on our observations and the graphs, it can be concluded that the ZELDA algorithm can provide superior energy efficiency in the setup and discovery phase when compared to the Data Funneling algorithm.

References

1. Dragan Petrovic et al, "Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks", IEEE Sensor Network Protocols and Applications (SNPA) 2003
2. J. Rabaey et al., “Pico Radio supports ad hoc ultra-low power wireless networking”, IEEE Computer, July 2000
3. Goldsmith and S. Wicker, “Design challenges for energy-constrained ad hoc wireless networks”, IEEE Wireless Communications Magazine, Aug 2002

4. R. C. Shah and J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks”, IEEE Wireless Communications and Networking Conference, March 2002
5. K. Toh, “Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks”, IEEE Comm. Mag., June2001
6. S. Singh et al, “Power aware routing in mobile ad hoc networks”, IEEE/ACM MobiCom, Oct. 1998
7. C. Intanagonwiwat et al, “Directed Diffusion: A scalable and robust communication paradigm for sensor networks”, IEEE/ ACM Mobicom, 2000
8. E. Royer, C. K. Toh, “A review of current routing protocols for ad hoc mobile wireless networks”, IEEE Personal Comm., April 1999
9. C. Perkins and E. Royer, “Ad hoc on demand distance vector routing”, Proceedings of 2nd IEEE Workshop Mobile Comp. Sys. & Apps. Feb 1999
10. D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks”, Mobile Computing, Kluwer, 1996
11. Chang and L. Tassiulas, “Energy conserving routing in wireless ad hoc networks”, IEEE Infocom, 2000
12. T. Cormen et al., Introduction to Algorithms, MIT Press, Cambridge, MA, 2002
13. Sedgewick, R. "Permutation Generation Methods", Computational Surveys 9, 137-164, 1977
14. C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges", Proceedings of the IEEE, vol. 91, no. 8, August 2003
15. Omnet++ Community Site, Available online[http://www.omnetpp.org]
16. Omnet++ Wiki, Omnet++ in a Nutshell, Available online [http://www.omnetpp.org/pmwiki/index.php?n=Main.OmnetppInNutshell]
17. S. Joseph Levine, “Writing and Presenting your Thesis or Dissertation”, Michigan State University, Lansing, MI, January 2005.
18. Edgar H. Callaway, Jr., “Wireless Sensor Networks: Architectures and Protocols”, ISBN 0-8493-1823-8.
19. Tarun Anand Malik, “Target Tracking in Wireless Sensor Networks”, A Thesis, Submitted for Master of Science in Engineering, Louisiana State University, May 2005
20. DeLynn Bettencourt, “A System of Performance Measurement of Pico Radio Network Protocols”, SUPERB 2002, College of Engineering, University of California, Berkeley
21. Lizhi Charlie Zhong, Rahul Shah, Chunlong Guo, Jan Rabaey, “An Ultra-Low Power and Distributed Access Protocol for Broadband Wireless Sensor Networks”, presented at Networld + Interop: IEEE Broadband Wireless Summit, Las Vegas, 2001. [http://bwrc.eecs.berkeley.edu]
22. Simulation Software Development Frameworks, Topology.org, available Online [http://www.topology.org/soft/sim.html]
23. Writer's Handbook, The Writing Center, University of Wisconsin Madison, Available Online[http://www.wisc.edu/writing/Handbook/index.html]
24. Kenneth A. Berman, Jerome L. Paul, Fundamentals of Sequential and Parallel Algorithms; Boston, MA, PWS Publishing Company
25. Yedidyah Langsam et. al., Data Structures using C and C++, 2nd Ed.
26. Deitel and Deitel, C++ How to program, 3rd Ed.
27. Silberscatz et al, Indexing Spatial Data(presentation),Database System Concepts, Available Online [www.db-book.com]
28. Wei Ye,John Heidemann, Deborah Estrin, an Energy Efficient MAC Protocol for Wireless Sensor Networks, Available Online [http://citeseer.ist.psu.edu/ye01energyefficient.html]
29. Jan Rabaey et al ,Pico Radio Communication/Computation Pico Nodes for Sensor Networks, DARPA Report 2001, EECS Department & Electronics Research Lab University of California at Berkeley
30. A.D. Amis, R. Prakash, T. H. P. Vuong and D. T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", in Proceedings of IEEE INFOCOM, March 2000.
31. L. Valiant. Universality Considerations in VLSI Circuits. IEEE Transactions on Computers, 1981.
32. T. Biedl and G. Kant. A Better Heuristic for Orthogonal Graph Drawing. In 2nd Annual European Symposium in Algorithms, 1994.
33. A. D. Amis, R. Prakash, T. H. P. Vuong and D. T. Huynh, “ Max-Min D-Cluster Formation in Wireless Ad Hoc Networks”, in Proceedings of IEEE INFOCOM, March 2000
34. A.D. Amis, and R. Prakash, “Load-Balancing Clusters in Wireless Ad Hoc Networks”, in Proceedings of ASSET 2000 , Richardson, Texas, March 2000
35. Gadyy Kozma, Zvi Lotker, et al., Geometrically Aware Communication in Random Wireless Networks (2004)

HAMID R MAJLESEIN
Dr. Majlesein currently serves as a Professor and Associate Dean for Research and Graduate Program of College of Engineering at Southern University and A&M College. His areas of interest are Electric Power Systems, Computer Networks and Digital Signal Processing. His teaching interests are in the areas of Network Analysis, Electric Machine, Power Systems, Computer Networks, Signal and Systems, and Digital Signal Processing.

SUJEET BHATTE
Mr. Sujeet Bhatte is currently a full time student at Southern University and A & M College. He has successfully passed his MS thesis and he will be graduating in Spring 2008. At Southern University he has worked with various on campus departments and organizations in areas related to computer networks and telecommunications. He came to the US after completing his Bachelor's Degree in Electronics Engineering with First class from Mumbai University, Mumbai India. Prior to coming to US he worked with CMS Computers Ltd as an Engineering Intern where he developed data communication equipment and tested electronic products.

ZHENGMAO YE
Dr. Ye currently serves as an Assistant Professor of Electrical Engineering at Southern University. Dr. Ye's research interests include modeling, control and optimization with diverse applications on automotive, electrical, mechanical and biomedical systems, as well as signal processing and image processing. Dr. Ye is a Senior Member of IEEE and the Founder and Director of System and Control Laboratory at Southern University.