An Integrated Framework for Wireless Sensor Network Management

by

Lutful Karim

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Doctor of Philosophy

in

Computer Science

Guelph, Ontario, Canada

Abstract

An Integrated Framework for Wireless Sensor Network Management

Lutful Karim
University of Guelph, 2012

Advisor:
Dr. Qusay H. Mahmoud

Wireless Sensor Networks (WSNs) have significant potential in many application domains, and are poised for growth in many markets ranging from agriculture and animal welfare to home and office automation. Although sensor network deployments have only begun to appear, the industry still awaits the maturing of this technology to realize its full benefits. The main constraints to large scale commercial adoption of sensor networks are the lack of available network management and control tools for determining the degree of data aggregation prior to transforming it into useful information, localizing the sensors accurately so that timely emergency actions can be taken at exact location, and scheduling data packets so that data are sent based on their priority and fairness. Moreover, due to the limited communication range of sensors, a large geographical area cannot be covered, which limits sensors application domain. Thus, we investigate a scalable and flexible WSN architecture that relies on multi-modal nodes equipped with IEEE 802.15.4 and IEEE 802.11 in order to use a Wi-Fi overlay as a seamless gateway to the Internet through WiMAX networks. We focus on network management approaches such as sensors localization, data scheduling, routing, and data aggregation for the WSN plane of this large scale multimodal network architecture and find that most existing approaches are not scalable, energy efficient, and fault tolerant. Thus, we introduce an efficient approach

for each of localization, data scheduling, routing, and data aggregation; and compare the performance of proposed approaches with existing ones in terms of network energy consumptions, localization error, end-to-end data transmission delay and packet delivery ratio. Simulation results, theoretical and statistical analysis show that each of these approaches outperforms the existing approaches. To the best of our knowledge, no integrated network management solution comprising efficient localization, data scheduling, routing, and data aggregation approaches exists in the literature for a large scale WSN. Hence, we efficiently integrate all network management components so that it can be used as a single network management solution for a large scale WSN, perform experimentations to evaluate the performance of the proposed framework, and validate the results through statistical analysis. Experimental results show that our proposed framework outperforms existing approaches in terms of localization energy consumptions, localization accuracy, network energy consumptions and end-to-end data transmission delay.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my advisor Dr. Qusay H. Mahmoud for his continuous support, enthusiasm, and encouragement. Under his supervision, I learned many important aspects in research, and overcame many difficulties, especially in writing a good thesis.

I would also like to express my sincere gratitude to my former advisor, Dr. Nidal Nasser, who helped me greatly in the research, writing and publishing results in scholarly journals and conferences. I am very grateful to Dr. David Swayne and Dr. Simon Yang for being on my advisory committee and for their support and valuable comments. I would also like to thank Dr. Stefan Kremer for chairing the defense, Dr. Kui Wu (external examiner) for his valuable feedback, and Dr. Mark Wineberg for his insights on using statistical analysis tools.

Last but not least, I would like to thank my parents, my wife, only daughter, brothers and sisters, friends, and all the people who have supported me along the way.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In recent years, Wireless Sensor Networks (WSNs) have achieved a widespread applicability in many areas such as environmental monitoring, health, agriculture, security, and battlefield since a WSN consists of a large number of tiny sensor nodes that can be deployed randomly, configured and controlled automatically without any human intervention. However, sensors are battery powered and thus, have limited energy. They are randomly deployed in human inaccessible remote, and harsh environments such as mountain, forest, ocean and can fail due to interference, natural calamity, and energy shortage. Thus, designing energy efficient and reliable network management approaches such as sensors localization, routing protocols, data scheduling and data aggregation approaches are very important in WSN research.

This chapter is organized as follows. Section 1.1 presents a typical WSN architecture with its applications, important research issues, and motivation of this research to design and implement a framework for WSN management and control. Section 1.2 presents the research statement of this thesis. Section 1.3 provides a summary of the contributions introduced in this thesis. Section 1.4 presents the organization of thesis and the name of refereed journals and conferences, where parts of this thesis have been published. Finally, a summary of this chapter is presented in Section 1.5.

1.1 Background and Motivation

A large quantity of the world's natural resources is still unexplored. For example, Africa has huge natural resources including diamonds, gold, oil and uranium, which are still undiscovered because manual exploration is very difficult in deep forest, remote, underdeveloped regions of Africa or other parts of the world. Even in many areas of developing world such as in the Northwest territories of Canada, monitoring health for elderly, resource exploration, monitoring the status of wildlife and wildlife habitat become difficult due to harsh environment/climate. Water, and forest resources are about 75% of the land in the Northwest territories, where natural resources include natural gas, petroleum, gold and diamonds. However, most areas are still unexplored for these valuable resources. Ensuring sustainability of about 33-million hectares forest use by preventing forest fire and through careful management is also a great challenge in the Northwest territories. Researchers (geologists, agronomists, environmental scientists) are working on different projects in these areas for exploration of natural resources. There are many existing technological solutions of these problems. However, most of the solutions are expensive and designed for a small scale specific application. Thus, a technological solution with the following characteristics is greatly required.

- Inexpensive so that it is affordable and can be used in developing countries of the world as well;
- Automatic control and configuration;
- Random deployment so that they can be used in areas, where manual deployment or intervention (in person) is not possible;
- Scalable so that they can be used in large geographical areas; and

- Easily and remotely accessible and controllable by end users.

A Wireless Sensor Network (WSN) can be the best technological solution with all above mentioned characteristics. A WSN consists of numerous unattended, resource-constrained, low power, and memory sensor nodes, which are of two types: sensor node, and sensor gateway or base station (BS). The sensor node has the basic capabilities of sensing, processing, and communicating. However, the sensor gateway or BS has more functionality besides these basic capabilities. It can collect, analyze, and process raw sensor's data and be connected to the Internet to share the data worldwide [23]. Based on the two types of sensors, a WSN normally constitutes a wireless ad hoc sensor network. Figure 1.1 illustrates a typical WSN, in which a sensor node senses different environmental parameters such as temperature, pressure, humidity, and sends data directly to the BS if the node is within the communication range of BS or through other nodes using multi-hop routing. Finally, data from BS reach end users through different communication networks, such as Internet.

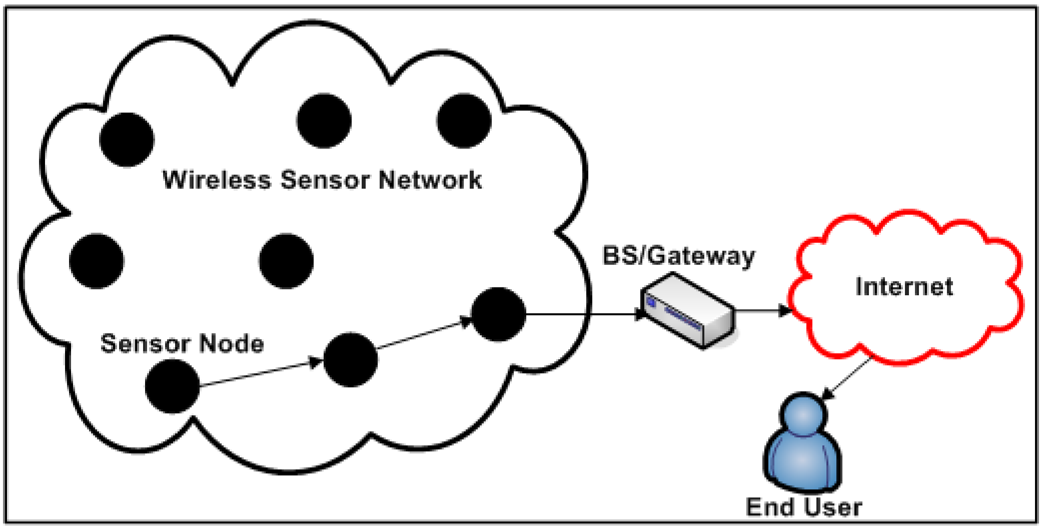

Figure 1.1: A Wireless Sensor Network

Although sensor network deployments have only begun to appear, the industry still awaits the maturing of this technology to realize its full benefits. Limited energy of sensors also makes it difficult to operate the network for a longer period of time. Hence, power consumptions of sensor nodes should be properly utilized, and reduced to prolong the lifetime of WSN. Most of the existing routing protocols of WSN deal with the energy efficiency of sensors. However, sensor nodes are also required to use a fault tolerant routing protocol to ensure continued network operation in the harsh, remote, and unattended environment such as deep forest, deserts, mountain, and Iceland. Hence, designing a fault tolerant, and energy efficient routing protocol that trades-off between the energy usage and reliability is an important research issue in WSN.

Due to the limited communication range of sensors, a large geographical area cannot be covered with inexpensive solution. Moreover, to relay data from fields to users through Internet, each BS needs to be connected to the Internet, which results in a large number of Internet subscriptions. In addition, as each sensor network has an individual Web interface, users do not have a complete or common view of different geographic sensor fields. Hence, a large scale sensor network with a common application interface is required. However, the main constraints to the large scale commercial adoption of sensor networks are the lack of efficient network management and control tools for determining the degree of data aggregation prior to transforming it into useful information, efficient routing, data packet scheduling, and localization techniques. Trade-off energy consumptions and end-to-end data transmission delay using limited or battery powered sensor devices is another challenge in a large scale deployment.

More specifically, existing sensor's localization approaches are not considered energy efficient and accurate for a large scale WSN for the following reasons:

1. They are mostly range-based, which require energy consuming range estimation techniques such as Time of Arrival (TOA), Time Difference of Arrival (TDOA),

and Angle of Arrival (AOA) [124, 100].

2. Some of these techniques use GPS, which are expensive to equip with all sensor nodes.

3. Most approaches require a large number of static anchor nodes.

4. These techniques consider a newly localized node as an anchor node, and result in a large localization error since error propagates from a newly localized node.

Similarly, existing routing, and data aggregation approaches are not energy efficient because they use flooding for path establishment. A large number of switching between aggregation levels requires a large number of control message transmissions. Moreover, these techniques do not provide fast data delivery because they use tree structure where nodes wait until they receive data from all children nodes at lower levels. Most existing localization, routing, data scheduling, and aggregation approaches are not scalable and fault tolerant. Finally, to best of my knowledge, no integrated framework consisting of localization, data scheduling, routing, and data aggregation exists for a large scale WSN application.

Thus, an integrated network management approach to overcome the aforementioned challenges for sensors' localization, topology control and routing protocols, data packet scheduling and aggregation techniques is required.

1.2 Research Statement

The focus of this research is to explore how to design and develop an efficient framework that integrates the network management, and control components (e.g., localization, routing, data scheduling and aggregation) for a large scale Wireless Sensor Network (WSN) in terms of energy consumptions, localization error, end-to-end data transmission delay, and packet delivery ratio.

More specifically, such a framework should have the the following properties.

1. Distributing nodes to an area in such a way that the network is fully covered or no sensing hole exists.

2. Localizing sensors through the use of local maps with reduced localization error.

3. Increasing the data delivery rate, and guaranteeing fast data delivery by designing efficient data scheduling, and aggregation approaches. This is because fast data delivery is very important and a major design challenge for a large scale WSN network that can be accessed remotely from a distant place.

4. Reducing energy consumptions by using a minimum number of active nodes, and data scheduling scheme.

5. Introducing fault tolerance since most of the existing network management components (e.g., routing protocols) are not fault tolerant.

Combining all these network management components into a framework in an efficient way that can trade-off energy efficiency, end-to-end data transmission delay, and data accuracy, is a great challenge. To the best of our knowledge, there is no WSN framework that integrates all these network management tools together.

1.3 Thesis Contributions

This research investigates a scalable and flexible Wireless Sensor Network (WSN) architecture that relies on multi-modal nodes equipped with the IEEE 802.15.4 and IEEE 802.11 and focuses on designing and implementing efficient network management approaches such as sensors localization, data scheduling, routing, and data aggregation for the WSN plane in this architecture and integrating them. The details of the WSN architecture and management approaches are presented in Chapter 3. This thesis introduces five main contributions in this context, which are as follows.

- **Range-free localization approach**: We design and implement an efficient range-free localization approach for a large scale WSN. This localization approach uses mobile anchor nodes which are less in number as compared to static anchor nodes that most existing approaches use. Moreover, this approach is scalable since mobile anchor nodes move to a large geographical area to localize sensor nodes whereas it is not the case for static anchor nodes. The proposed localization approach uses sensing range instead of communication range of sensors. This phenomenon increases localization accuracy, and reduces energy consumptions. This is because messages with anchor's position information are transmitted to un-localized nodes at shorter distance (sensing range), and the region of three intersected sensing circles of anchor nodes where the un-localized node reside is very small.

- **Efficient topological structure and routing protocol**: The proposed Cluster-based Routing and Topology Control Approach (CRTCA) selects a minimum number of active nodes in each zone that participate in routing by creating the shortest data transmission path from a source node to BS. The number of active nodes are selected in such a way that reduces the probability of having sensing hole in the network. This routing approach balances energy consumptions by properly distributing nodes to different zones of the network, such as, zones that are closest to BS will have a few more nodes than the zones that are far away from BS. This is because nodes in the farthest zones transmit data through the nodes in zones that are closed to BS.

  This routing approach achieves fault tolerance by introducing subscription/publish paradigm, small-sized special packets, and alternative paths. During path creation, an alternative node or path is selected for each active node A on the data transmission path, which works as an active node whenever A fails. BS subscribes to nodes for some event of interests. If a node, X senses such an event, X sends the data to its neighboring active node on the data transmission path or BS. Otherwise, X sends a small-sized special packet just to notify the neighboring node or BS that it is still alive. If X does not send any data or special packet at its allocated timeslot BS understands that X has failed. Then data of X will be transmitted through an alternative node or path.

- **Priority-based packet scheduling approach**: We design and implement a priority-based packet scheduling scheme that reduces end-to-end data transmission delay by allocating maximum three levels in the ready queue of each node. For instance, each leaf node will have two levels in its ready queue to hold real-time and non-real-time data. Upper level nodes will have three levels in the ready queue to temporarily store real-time, non-real-time remote data, and non-real-time local data. This approach schedules data based on their priority. For instance, real-time data have the highest priority and incur less end-to-end data transmission delay. This is one of the main objectives of using WSN in emergency applications. However, this scheduling approach achieves fairness by allowing the lowest priority non-real-time local data (i.e., data that are sensed at the current sensor node) to be transmitted after a certain number of timeslots if they cannot be transmitted due to the continuous arrival of the higher priority non-real-time remote data (i.e., non-real-time data, a node receives for the lower level nodes).

- **Dynamic data aggregation approach**: We introduce an energy efficient, and dynamic data aggregation approach that allocates a minimum number of aggregating nodes in the network. This approach is dynamic since aggregation functions are changeable or adjustable based on the data types. For instance, real-time data will be transmitted without any aggregation, and non-real-time data will be aggregated at intermediate nodes based on the aggregation functions, MIN, MAX, MEAN etc. Moreover, this approach reduces the end-to-end data transmission delay using Time Division Multiplexing (TDMA) scheme where, nodes at different levels have variable length timeslots. For instance, the duration of timeslot at the lowest level L_k will be less than that at the upper level L_{k-1}.

- **Integrated network management framework**: We efficiently integrate the proposed localization, data scheduling, routing, and data aggregation approaches in a complete single framework so that it can be used in a large scale WSN application to reduce the overall network energy consumptions, and data transmission delay. Most existing WSN management frameworks do not include all these components together. For instance, an existing framework proposes a routing protocol with the assumption that sensors know their locations.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 presents literature on different network management components of a Wireless Sensor Network (WSN) such as localization methods of sensor nodes, packet scheduling approaches, routing protocols, and data aggregation approaches of WSN with their limitations, classification, and comparison. Chapter 3 presents the multimodal wireless network architecture having WSN, Wi-Fi, and WiMAX networks, and the detailed architecture of the

WSN plane with the working flow of different components of WSN. We also present assumptions and define important terminologies and performance metrics in Chapter 3. Chapters 4 and 5 present the proposed sensor's localization approach, and routing protocol, respectively. Chapters 6 and 7 present the proposed packet scheduling and data aggregation approaches, respectively, with their performance analysis and results. In Chapter 8 we evaluate the performance of integrated network management framework. Conclusions and ideas for future work are discussed in Chapter 9.

Parts of this thesis have been published in refereed journals and conferences. Our proposed localization approach, namely, Range-free and Energy efficient Localization using Mobile Anchor node (RELMA) has been published in the IEEE Global Communications Conference (GLOBECOM 2010) [49]. We have also designed and implemented an energy efficient, and fault tolerant dynamic clustering protocol for WSN, which has been published in the IEEE GLOBECOM 2009 conference. In this protocol, base station (BS) subscribes to all sensor nodes in the network for some event of interests. If member nodes sense any event of interest it sends that data to cluster head (CH). Otherwise, they send a small sized special packet to CH to notify that they are still alive. However, this protocol consumes much energy for sending redundant data by member nodes of a cluster that reside close to each other. Thus, we introduce a zone-based routing protocol, namely, Cluster-based Routing and Topology Control Approach (CRTCA) [47] that has been published in the 7th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2011). The CRTCA protocol selects the minimum number of active member nodes of a zone or cluster in such a way that also reduces the possibility of having any sensing hole. Moreover, the number of redundant data sensing is reduced. Afterwards, we introduce a new set of graphs referred to as the Mini Gabriel (MG) graphs to the CRTCA protocol that outperforms the existing Gabriel

Graph (GG) [81, 29] in terms of total network energy consumptions, routing delay, and data transmission delay. This work has been published in the EURASIP Journal on Wireless Communications and Networking [51]. Moreover, designing efficient packet scheduling, and data aggregation approaches are very important for reducing the end-to-end data transmission delay of the network. Thus, we have introduced a priority-based packet scheduling algorithm that has been accepted to be published in the IEEE International Conference on Communications (ICC 2012) [54], and a dynamic data aggregation approach that has been published in the IEEE Global Communications Conference (GLOBECOM 2010) [48].

1.5 Summary

In this chapter, we presented the applications of a Wireless Sensor Network (WSN) with its important research issues, and motivations to design and implement a WSN management and control framework that comprises sensors localization, routing, data scheduling, and data aggregation approaches. We also presented contributions of this thesis and mentioned parts of the contributions that have been published in refereed journals and conferences. In the next chapter, we will present existing localization, routing, data scheduling, and data aggregation approaches in the literature.

Chapter 2: Literature Review

Wireless Sensor Networks (WSNs) are used in environmental monitoring, air pollution detections and control, agriculture, health and many other applications by deploying sensors that gather information of different natural phenomenon such as temperature, light, humidity, pressure, and soil moisture from surroundings. Sensors can be deployed randomly, configured and controlled automatically, accessed and controlled remotely, which are considered as unique features of WSN. These features also allow WSNs to be used for unattended applications in deep forest, underwater, and harsh environments. However, sensors have limited energy, and other resources (e.g., memory, and processing capabilities). Thus, designing energy efficient and reliable WSN management approaches such as sensors localization, packet scheduling, routing protocols, and data aggregation by identifying their limitations in the existing systems are very important.

In this chapter, Sections 2.1, 2.2, 2.3, and 2.4 present the literature on localization approaches, packet scheduling approaches, routing protocols, and data aggregation approaches of WSN with their limitations, classifications, and comparisons, respectively.

2.1 Sensors Localization

Locating the position of a sensor node in the network has a great importance in most WSN applications where sensors are deployed randomly in remote, and harsh environments. Without location information sensor data is meaningless. For instance, identifying the locations of sensors that sense the temperature, and gas (CO, CO_2) concentrations are required to detect the location of wildfire. Similarly, localizations are used to identify the location of intruder,i.e., enemies in battlefield, find the optimum coverage of a network and optimal routes in geographic routing [10]. Existing localization approaches use Global Positioning Systems (GPS) in sensor nodes which is expensive to equip with all nodes. To reduce the cost, a few sensor nodes are equipped with GPS. These sensor nodes are known as anchors nodes. The anchor nodes broadcast their positional information through beacons. An un-localized node can estimate its position whenever it receives a number of beacons. However, GPS do not function for indoor and many outdoor WSN applications since there is no straight line of sight (LOS) from nodes to satellites. There are some other techniques that use the interconnectivity of nodes instead of using a large number of anchor nodes to estimate their positions. In the following sections, we present position estimation techniques for blind nodes and different types of localization approaches.

2.1.1 Position Estimation Techniques

In this section, we present the position estimation techniques, Lateration, Angulation, Probabilistic approach, and Fingerprinting approach [124].

Lateration calculates the coordinate of a blind node based on the precise coordinate of three anchor nodes. Trilateration determines the intersection point of three circles where the center and radius of each circle are the coordinate, and sensing radius of an anchor node, respectively. Hence, this method requires three anchor nodes and multilateration requires more than three anchor nodes. Angulation or triangulation calculates coordinates based on the information about angles instead of distances. This method determines the position of an unknown node x by measuring angles to it from two anchor nodes that forms a triangle connecting two anchors and x with one known side (between two anchors) and two known angles (between each anchor and x). Probabilistic approach is a statistical approach, where errors in distance estimations are modeled as normal random variables for finding the probable locations of the blind nodes. This approach uses the collective information received from the neighboring nodes. Fingerprinting technique is a Received Signal Strength (RSS) [124] value based pattern matching problem, which creates a database of pre-measured RSS value for each sensor location. Then, this approach localizes a node x by matching the RSS value of x with a RSS value in the database. In the following sections we present existing localization approaches with their classifications, comparison, and discussion.

2.1.2 Centralized and Distributed Localization Approaches

In centralized localization approaches, nodes send their location information to a central server or base station (BS) that localizes other nodes. Area localization scheme (ALS) [76] is an example of a centralized localization method that estimates the position of sensor nodes in an area rather than their exact location. This method uses anchor nodes, normal sensor nodes, and sinks. Anchor nodes send beacon messages to sensor nodes to localize themselves. Sensor nodes measure the power level obtained from the anchor nodes and forward the information with other sensors data to the sink that has a high computational power to perform localization. The work done by Chaurasiya V.K. et al. [20] proposed a centralized localization algorithm using directional antenna, where each node collects information from other nodes and sends to the central server to estimate its coordinate. In this approach, an anchor node initially sends its location information (on each directional antenna with an antenna ID) to a neighboring un-located node x. Then x localizes itself based on the receipt information from anchor node such as distance and angle of arrival and becomes an anchor node. Similarly, x sends its information to neighboring nodes to localize them. The estimated location of the un-located node is calculated as $(x_n, y_n) = (x_a + d_i \cos\theta_j, y_a + d_i \sin\theta_j)$ where d = distance between the anchor and unknown node, θ = angle between the anchor and unknown node, d_e = error in distance calculations, θ_e = error in angle calculations, $d - d_e \leq d_i \leq d + d_e$, $\theta - \theta_e \leq \theta_j \leq \theta + \theta_e$. However, this approach does not work for mobile sensor nodes. Complexity increases for a large number of sensor nodes and a small beam width with directional antennas that is required for high localization accuracy.

In decentralized or distributed localization methods, each sensor node can localize itself, where a small number of communications take place between neighboring nodes. Localization is performed using reference/anchor nodes or self localization schemes. Distributed approaches are of two types: Range-based and Range-free, which will be discussed in the next section. Distributed localization methods have lower accuracy but they require lower number of communications than that in the centralized methods. Thus, these methods are energy efficient, and suitable for resource constraint sensor nodes.

2.1.3 Range-based and Range free Localization Techniques

Range-based localization techniques require extra processing or hardware to calculate distances and/or angles of the transmitting signals between nodes. These algorithms determine the distance between two nodes using Time of Arrival (ToA) [124, 100], Time Difference of Arrival (TDoA) [100], Angle of Arrival (AoA) [100], and Received Signal Strength Indicator (RSSI) [124] mechanisms. ToA measures the range through signal arrival time and has a high precision. However, this approach requires a high processing speed and time synchronization ability of sensor nodes. TDoA measures distance between two nodes by computing time difference of arrivals of two different signals such as radio frequency (RF) and ultrasound signals. Time difference is then translated into distance using the transmission speed of two signals. For instance, in Figure 2.1 the un-localized node P receives signals with three anchor nodes A_1, A_2 and A_3 at time time t_{A_1}, t_{A_2} and t_{A_3}, respectively having the speed S_c. Thus, distance D_1, D_2 and D_3 between P and each of A_1, A_2 and A_3 can be calculated by multiplying t_{A_1}, t_{A_2} and t_{A_3} with S_c. Based on these distances the circles are constructed.

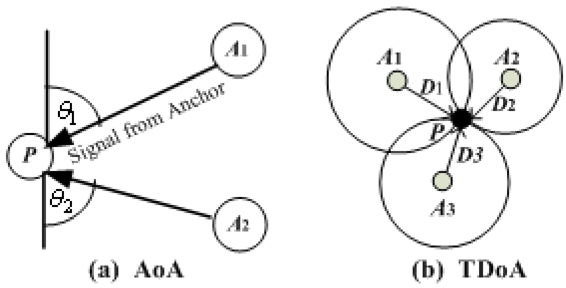

Figure 2.1: (a) AoA (b) TDoA range estimation techniques

However, TDoA relies on energy consuming extra processing and is not suitable for resource constraint sensor networks. Moreover, ultrasound signals require dense deployment of anchor nodes since ultrasound signals usually propagate very short distances, only a few meters. RSSI estimates the distance between two nodes based on the energy level of transmitted and received signals. However, this approach suffers from an inaccurate localization due to multipath fading of signals. AoA estimates the position by measuring angles between a target and anchor node. However, AoA methods use angle information from anchor nodes and ignore the interconnectivity among the neighboring blind nodes. Figure 2.1 illustrates AoA method.

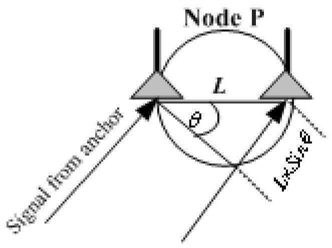

Figure 2.2: Cooperative AoA

The work done by Xu J. et al. [114] proposed a new cooperative AoA positioning algorithm that takes advantage of high connectivity of the sensor networks as well as the high data rate and fine time resolution of Ultra Wide Band (UWB) signals and identify the location of a sensor node by employing all neighboring nodes to achieve higher accuracy. In this approach, AoA is obtained by calculating the phase difference of the arrival signals and transforming the phase differences into AoA data. TDoA requires synchronization among all nodes that is not practical in sensor networks. Hence, Cooperative AoA uses the high interconnectivity of the networks instead of time synchronization. Moreover, the minimum number of anchor nodes required to locate a blind node in a 2-D positioning is only 2, which is less than that of other positioning schemes. Figure 2.2 demonstrates the estimation of angle θ in Cooperative AoA whenever, the un-localized P has a uniform linear array of antenna. The distance between signals at two receiving antennas is L. Thus, $\theta = \arccos(\frac{S_p \times \gamma}{L})$, where, S_p and γ are signal/light speed and time delay between neighboring antennas.

If the number of anchors is represented by $A(n)$, Range-based localization algorithms cannot accurately localize a sensor node if $A(n) < 3$. Zhang L. and Deng B. [124] propose a distributed range-based localization algorithms using genetic algorithms that can localize a node even if $A(n) < 3$. Initial location of an un-localized node P is obtained from a neighboring anchor node using the RSS value of incoming radio signals. Then if P has $A(n) \geq 3$, P calculates its position using trilateration [62]. Otherwise, P estimates its initial position from anchor nodes and refines using genetic algorithm. If $A(n) = 2$, P draws samples in the intersected region of the communication circles of two anchors. If $A(n) = 1$, P uses its only one neighboring anchor and anchors in two hops away.

Range free localization methods [36, 35, 83, 121, 95] make use of the properties of communication and sensing range between nodes to estimate the positions of sensor nodes instead of using additional hardware for range estimation. Though range-based algorithms are more precise than range free algorithms, range free algorithms are more energy efficient and are used for applications where the precision of sensors are not so much important. Range free algorithms are of two types: (i) Incremental, and (ii) Concurrent. Incremental algorithms initially assign coordinates to a set, S of three or four nodes. Then these algorithms repeatedly calculate coordinates of other un-located nodes by measuring distances to nodes whose coordinates were computed previously, and add to S. However, error in a localized node is propagated to localize other nodes that results in a poor coordinate assignment. In Concurrent algorithms, all nodes calculate their positions simultaneously. Some of these algorithms use iterative optimization to reduce differences between measured and calculated distances. These algorithms work by dividing networks into sub-regions that might have a small overlap and create local maps (by assigning coordinate) and append all local maps into a global map.

Centroid [36] is range-free algorithm that uses centroid of a polygon as the estimated position an un-located node X. This polygon is constructed by connecting the coordinates of all neighboring anchor nodes of X. However, Centroid requires a large number of anchor nodes. Point-In-Triangulation Test (PIT) [121, 36, 83] is also a range-free localization technique, where a node X chooses three anchor nodes from which it receives a beacon message and checks whether X is inside a triangle that is formed by connecting the coordinates of these three anchor nodes. Then Adaptive PIT (APIT) [36, 83] localization technique is used to estimate the position of X using Center of Gravity (COG) that is the intersected area of all triangles in which X resides. Initially, a node receives beacon from n anchors and forms all possible triangles connecting these beacon locations. The triangles that pass the PIT test and whose intersection forms a COG are used to estimate the position of X. Figure 2.3 shows that node X resides in the colored area that is the intersected area of all triangles.

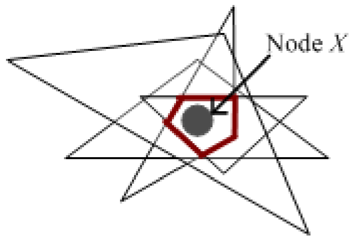

Figure 2.3: APIT localization approach

APIT uses a centralized as well as distributed approach as information of each node is transmitted to a central node for final computation. APIT requires a high density of anchor node to be effective. Coordinate [10, 83] is a range free algorithm which does not use anchor nodes. Each node makes itself a local reference node by making itself a coordinate origin, communicates and calculates distances among neighboring nodes. Thus, the number of reference nodes increases. Amorphous algorithm [10, 83] is a range free multi-hop localization approach. In this method, each sensor node obtains the coordinate and hop count information of at least three anchor nodes and forwards to its neighboring nodes. A node discards a message if the hop count value of this message is larger than the hop count value of an already received message. Sensors store these values and compute distances from anchors by selecting the lowest hop count received.

DV-hop [37, 83] is a range free algorithm, where anchor nodes broadcast packets that contain their positional information and a flag that tells the number of hops anchor nodes are away from the current receiving node. Flag (hop number) is initialized to one and increased by one, when a packet from an anchor node passes through a relay node. Hence, each node knows the hop distance from anchor nodes. Thus, Average Distance per Hop (ADH) is calculated and broadcasted. Then each node can calculate its distance to anchor nodes by multiplying ADH with the number of hops. Figure 2.4 shows the hop count (HC) value of each node B, C, D, E from the anchor node A. ADH based on A can be easily calculated. However, calculating the distance of a node to an anchor node using ADH might have substantial error. For instance, though the node E is more than one hop away from node A it is very close to one hop using a straight line (if drawn from A). Thus, multiplying the ADH with HC value results in a large error in euclidian distance.

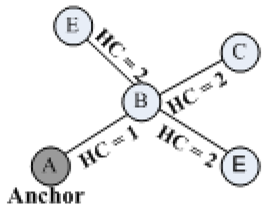

Figure 2.4: DV-hop localization approach

ADH also varies for different anchor nodes and hence, distances from an un-located node to anchor nodes might have substantial errors. Moreover, in DV-Hop algorithm, the anchor nodes can influence calculating the location of un-localized nodes. For instance, the nearer the distance between the anchor nodes and un-localized nodes, the higher the accuracy of location is and vice versa. Thus, DV-Hop algorithm is suitable only for dense WSNs. Improvements in DV-hop localization approaches are done in [125, 75, 32] to eliminate the above mentioned problems. The work done by Lu Q. et al. [75] proposes NDV-hop localization algorithm, where an unknown node calculates its location based on the distance between itself and all other anchor nodes. Thus, the localization accuracy will be more in both sparse and dense WSNs. The work done by Gui, L. et al. [32] proposes another improved DV-hop, namely, Selective 3-Anchor DV-hop algorithm. In this approach, any three anchors can form a 3-anchor group. Based on each group, the un-localized node generates a 3-anchor estimated position. Among all the 3-anchor estimated positions the one which has the smallest connectivity difference in terms of hops with the un-localized node is selected as the final estimated position.

In general, in DV-hop algorithms, localization errors will be propagated since a newly localized node acts as an anchor node. Thus, error in a newly localized node increases incorrect localizations in other un-located nodes. To minimize errors in DV-hop, Tian S. et al. [100] proposed Selective Anchor Node Localization Algorithm (SANLA) that selects three most accurate anchor nodes to estimate the distance of un-located nodes using trilateration. In SANLA, un-located nodes get anchor lists and ADH using DV-hop. Then an un-located node x chooses two anchor nodes out of all anchors in its list. Each time x chooses a reference node y from which x received ADH first and localizes itself using trilateration. Then x becomes a reference node and y becomes an unknown node. The same process is repeated that minimizes error in identifying the final position of x. However, finding three accurate anchor nodes in SANLA increases the processing cost. Hence, the work done by Shanliang Li et al. [66] proposed Neighbor-information-Based Localization System (NBLS) that uses both hop count in DV-hop and neighboring nodes information. To estimate the initial position of an un-located node a_1 with coordinate (x_1, y_1) Node Distance (ND) of a_1 with each of its neighboring anchor nodes a_2, a_3, a_4 are calculated as an ratio of the total number of neighboring nodes of a_1 to the number nodes which are neighbors of both a_1 and an anchor node. Then using the ND and coordinates of the anchor nodes the position of a_1 is calculated. In the first phase of the algorithm each node sends a message including its neighbor and hop count information to BS (centralized method) that is originated at an anchor node. BS can estimate the coordinate of the unknown node that sends three or more messages. However, NBLS suffers from collision of messages that are transmitted from a node to BS. The work done by Yuanhua H. and Hongsheng L. [37] proposed a distributed algorithm based on a believable factor namely, wbf-Euclidean. In some existing approaches, errors are propagated since they consider a newly localized node as an anchor node. Hence, a believable factor (threshold) is used to consider a newly localized node as an anchor node only when the location error is small enough. This error is computed based on the estimated coordinate of the unknown node x using an anchor node y and measured distance between x and y. The anchor node y has a believable factor and a threshold (a constant value) associated with it. If the believable factor is greater than the threshold only then x is localized and becomes an anchor node. For initial anchor nodes, believable factor is set to a larger value and whenever an unknown node becomes an anchor node its believable factor is associated with and directly proportional to the location error. Hence, if the location error is high, the believable factor will be small.

Among other localization approaches, in Cluster-based Iterative GPS-free Localization algorithm (CIGL) [22], a group of nodes in the network forms Local Coordinate Systems (LCSs) using clustering approach. Then all the LCSs converge to the global coordinate system by rotation and reflection transformation and locate the nodes. Finally, the remaining un-localized nodes are iteratively localized using the newly localized nodes, which are considered as new beacon nodes. In [14], a range-free localization algorithm, LLSiWSN is proposed for a large scale WSN. LLSiWSN uses either a few or no anchor nodes. In [64] Lee S. and Kim K. propose a selection method of the communication range since the communication range places a vital role in localization accuracy. It has been observed that the average localization error is high when the average node density is low and vice versa. Thus, to make the local node's density higher and localization error lower, the communication range should be larger. In the next subsection, we present localization methods that use mobile anchor nodes.

2.1.4 Localization Methods using Mobile Beacons

When anchors are stationary, a large number of anchors are necessary since each un-localized sensor node collects location information from neighboring anchor nodes. To alleviate this problem mobile anchor nodes are used, where a few anchors nodes can transmit location information to all un-localized sensor nodes. Hence, these approaches are energy efficient but the environmental noise affects the localization performance. Directional antennas in sensors have a long transmission range and can counteract interference since nodes can receive signals only from the desired directions and avoid signals from unwanted directions.

The work done by Fengqi B.Z. and Zhang Y.Z. [121, 122] proposed distributed, energy efficient, reliable, accurate, and large scale, range free localization algorithm known as Border Line Intersection Localization (BLI) that employs a mobile anchor node. The mobile anchor node is equipped with a directional antenna that transmits and receives data only in one direction and a GPS receiver to get its location coordinate. Sensor nodes are equipped with an Omni-directional antenna that transmits and receives data in all directions and has less gain. A sensor node x receives the first and last beacon message from the mobile anchor node. When x finds two beacon points, it looks for the border lines for each beacon point. The intersection point of the border lines of two beacon points is the location of x. BLI can be used for sensor networks of any size and density and all computations are done locally since a single mobile beacon is used.

Wu Y.H. and Chen W.M. [111] proposed a simple Rectangle Overlapping Approach (ROA) where the mobile beacon is equipped with a GPS and a directional antenna. The mobile beacon moves along a predefined route and periodically broadcasts its position in the surrounding area so that sensor nodes in this area can estimate their positions. Initially, the moving beacon broadcasts the radius of its sensing range and rotation angle to all sensor nodes. Then the beacon moves a fixed distance and turns n times periodically with an angle θ and transmits signals through its directional antenna. At each turn it broadcasts the turning number and coordinates (x_b, y_b). When an un-localized node x receives the turning value i, x knows that its position is located in the sector area between $(i-1)\theta$ and $i\theta$. For example, if $\theta = 30$ degrees and x receives $i = 3$, then x knows its position is located in the sector area between 60 and 90 degrees.

Due to multi-path fading, environmental noise, and other related problems most distributed, range free, and $RSSI$-based localization algorithms cannot accurately find a beacon point for estimating locations. To alleviate this problem, Zhu Y. et al. [130] proposed a RSSI-based localization algorithm that uses a single mobile anchor. In this algorithm, an un-localized sensor node can localize itself by searching the expected position of beacon points, i.e., the maximum RSSI points. The first and second beacon points are collected when the mobile anchor passes through an un-localized node for the first and second time, respectively. The un-localized node computes its location using the two beacon points and works as a stationary anchor node. The localization algorithms presented so far use mobile anchor nodes for static sensor networks where all other nodes are stationary. Thuse, these algorithms do not work for Mobile Wireless Sensor Networks (MWSN). The work done by Kim K. and Hong Y. [59] propose a self localization algorithm for MWSN, where relay nodes are selected on the moving directions of anchor nodes that collect the position information of anchor nodes and send to un-located sensor nodes. In existing algorithms if an un-localized node has at least three anchor nodes in its transmission range/vicinity it can estimate its coordinate using triangulation. However, it might be difficult to have three anchor nodes in the communication range of an un-located node of MWSN since all nodes are moving. Hence, the proposed approach uses a minimum number of mobile anchors and a few relay nodes so that an un-localized node can communicate with an anchor at multiple-hops through relay nodes. To localize a sensor node only the selected relay nodes within the communication range of that sensor node can send anchors' positional information to minimize energy consumptions. The most optimal relay node is one that has the highest projected distance with the anchor node or maintains the longest proximity with the anchor node. A sensor node localizes itself using trilateration when it receives positional information from three relay nodes. However, this approach does not consider sensors of different types and capacity.

2.1.5 Discussion

Table 2.1: Comparison of different range-based localization approaches of Wireless Sensor Network

Features	Co-operative AOA	ROA	Self-Localization	Genetic Algorithm
Distributed Algorithm	√	√	√	√
Localization Parameter	AOA, TDOA	Angle of anti-clock wise rotation	RSSI	Trilateration
No. of Anchors	Min. 2	Low	3 or more	Low
Static Anchor	√	X	X	√
Mobile Anchor	X	√	√	X
Scalable	X	X	√	√
Static WSN	√	√	√	√
Mobile WSN	X	√	√	X

We find that most of the localization approaches are range-based, which use expensive hardware for the initial distance measurements using RSSI, TOA, TDOA, AOA, etc. Moreover, most of these approaches are distributed that collaboratively estimate the positions of nodes to increase the localization accuracy. These approaches initially use a few anchor nodes, which are incremented by assigning a newly localized node as an anchor node. Thus, a small localization error in the newly localized node is propagated to the un-localized nodes that results in a large localization error when a large number of nodes are localized. Table 2.1, and Table 2.2 compares the existing range-based localization techniques and range-free localization techniques on some important features, respectively.

Table 2.2: Comparison of different range free localization approaches of Wireless Sensor Network

Features	DV-Hop	APIT	Wbf-Euclidian	SANLA	BLI	NBLS
Distributed Algorithm	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	X
Centralized	X	$\surd$	X	X	X	$\surd$
Iterative	X	$\surd$	X	X	X	$\surd$
Concurrent	$\surd$	X	$\surd$	$\surd$	$\surd$	$\surd$
Localization Parameter	ADH	COG	Trilateration, Weighted Mean	ADH, Trilateration	Intersect point of 2 beaconlines	Hop count
No. of Anchors	High	High	High	Low (3)	Low (1)	Low (3)
Static Anchor	$\surd$	$\surd$	$\surd$	$\surd$	X	$\surd$
Mobile Anchor	X	X	X	X	$\surd$	X
Scalable	X	X	X	X	X	X
Static WSN	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Mobile WSN	X	X	X	X	$\surd$	X

2.2 Packet Scheduling

Scheduling data packets at sensor nodes, interchangeably used as task scheduling in this chapter, are significantly important since it determines the data transmissions order based on different factors such as data size, transmission deadline, and data priority. For instance, data sensed for real-time applications have higher priority than data sensed for non-real-time applications. Packet scheduling is very useful for heterogeneous Wireless Sensor Networks (WSNs) containing different types of sensors that sense different types of data. Existing research mostly focus on sensor's sleep and wake-up scheduling [4, 5, 13, 15, 16, 33, 71, 72, 94, 92, 97, 109, 108, 112, 116, 128, 43, 58, 60], link scheduling scheme [120], which is used under the physical interference model for Wireless Multimedia Sensor Network (WMSN), and TDMA scheduling for dynamic traffic patterns [126, 104]. However, only a few studies exist in the literature on the packet scheduling [86, 74, 118, 127, 27] of sensor nodes that schedule the processing of data events available at a sensor node and also reduces energy consumptions. In the following subsections, we classify packet scheduling schemes of WSN based on different factors (Figure 2.5), and present existing scheduling schemes of these types.

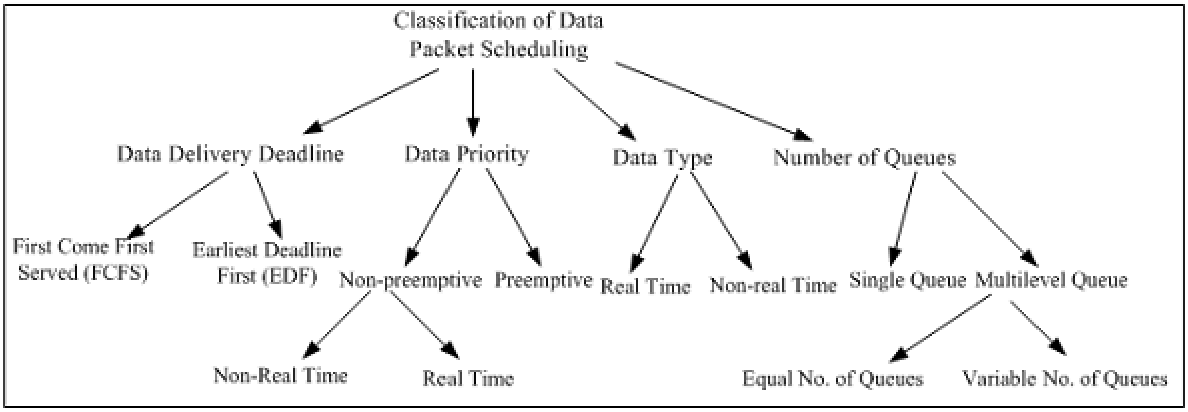

Figure 2.5: Classification of packet scheduling schemes

2.2.1 Classification Factor: Deadline

Packet scheduling schemes can be classified based on the deadline of arrival of data packets to the base station (BS), which are as follows.

First Come First Served (FCFS)

Most existing WSN applications use First Come First Served (FCFS) schedulers that process data in the order of their arrival times at the ready queue. In FCFS, data that arrive late at the intermediate nodes of the network from the distant leaf nodes require a lot of time to be delivered to base station (BS) but data from nearby neighboring nodes take less time to be processed at the intermediate nodes. In FCFS, many data packets arrive late and thus, experience long waiting times.

Earliest Deadline First (EDF)

Whenever a number of data packets are available at the ready queue and each packet has a deadline within which it should be sent to BS, the data packet which has the earliest deadline is sent first. This algorithm is considered to be efficient in terms of average packet waiting time and end-to-end delay.

The research work done by Lu C. et al. in [74] proposes a real-time communication architecture for large scale sensor networks, whereby they use a priority-based scheduler. Data that travel the longest distance from the source node to BS but have the shortest deadline is prioritized. If the deadline of a particular data packet expires, the relevant data packets are dropped at an intermediate node. Though this approach reduces network traffic and data processing overhead, it is not efficient since it consumes resources such as memory, computation power and increases processing delay. The performance of the scheme can be improved by incorporating FCFS. The work done by Kambiz Mizanian et al. [87] propose RACE - a packet scheduling policy and routing algorithm for real-time large scale sensor networks that uses loop free Bellman-Ford algorithm to find paths with the minimum traffic load and delay between source and destination. Since FCFS scheduling does not work well in real-time networks due to the fact that packets have different end-to-end deadlines, RACE uses the Earliest Deadline First (EDF) scheduling concept to send packets with earliest deadline. It also uses a prioritized MAC protocol that modifies the initial wait time after the channel becomes idle and the back-off window increasing function of IEEE 802.11 standard. Priority queues actively drop packets whose deadlines have expired to avoid wasting the network resources. However, local prioritization at each individual node in RACE is not sufficient because packets from different senders can compete against each other for a shared radio communication channel.

2.2.2 Classification Factor: Priority

Packet scheduling schemes can be classified based on the priority of data packets that are sensed at different sensor nodes.

Non-preemptive

In non-preemptive priority packet scheduling, when a packet t_1 starts execution, task t_1 carries on even if a higher priority packet t_2 than the currently running packet t_1 arrives at the ready queue. Thus t_2 has to wait in the ready queue until the execution of t_1 is complete.

Preemptive

In preemptive priority packet scheduling, higher priority packets are processed first and can preempt lower priority packets by saving the context of lower priority packets if they are already running.

Min Y.U. et al. [118] present packet scheduling mechanisms that are used in TinyOS [65, 101] - the widely used operating system of WSNs and classified them as either non-preemptive (co-operative) or preemptive. Co-operative scheduling schemes can be based on a dynamic priority scheduling mechanism, such as EDF and Adaptive Double Ring Scheduling (ADRS) [69], that uses two queues with different priorities. The scheduler dynamically switches between the two queues based on the deadline of newly arrived packets. If the deadlines of two packets are different, the shorter deadline packet would be placed into the higher priority queue and the longer deadline task would be placed into the lower priority one. Cooperative schedulers in TinyOS are suitable for applications with limited system resources and no hard real-time requirements. On the other hand, preemptive scheduling can be based on Emergency Task First Rate Monotonic (EF-RM) scheme. EF-RM is an extension of Rate monotonic (RM), a static priority scheduling, where the shortest deadline packet has the highest priority. EF-RM divides packets of WSN into periodic packets (PP) whose priorities are decided by a RM algorithm and non-periodic packets that have higher priority than PPs and can interrupt, whenever required, a running PP.

2.2.3 Classification Factor: Packet Type

Packet scheduling schemes can be classified based on the types of data packets, which are as follows.

Real-time packet scheduling

Packets at sensor nodes should be scheduled based on their types and priorities. Real-time data packets are considered as the highest priority packets among all data packets in the ready queue. Hence, they are processed with the highest priority and delivered to the BS with a minimum possible end-to-end delay.

Non-real-time packet scheduling

Non-real time packets have lower priority than real-time tasks. They are hence delivered to BS either using first come first serve or shortest job first basis when no real-time packet exists at the ready queue of a sensor node. These packets can be intuitively preempted by real-time packets.

Though packet scheduling mechanisms of TinyOS are simple and are used extensively in sensor nodes, they cannot be applied to all applications: due to the long execution time of certain data packets, real-time packets might be placed into starvation. Moreover, the data queue can be filled up very quickly if local data packets are more frequent that causes the discard of real-time packets from other nodes. To eliminate these drawbacks, Zhao Y. et al. [127] proposed an improved priority-based soft real-time packet scheduling algorithm. Schedulers traverse the waiting queue for the data packets and choose the smallest packet ID as the highest priority to execute. Each packet is assigned an Execute Counter, EXECUTE_MAX_TIME, i.e., the largest initial task execution time. The management component compares the current packet ID with the previous packet ID. If it is the same, the system executes it and decrements the counting variable. Otherwise, if the counting variable is null, the management component terminates this packet and other packets get the opportunity to be executed. However, packet priorities are decided during the compilation phase, which cannot be changed during the execution time. If high priority packets are always in execution, the low priority packets cannot be implemented. If low-priority packets occupy the resources for a long time, the subsequent high-priority packets cannot get response in time.

2.2.4 Classification Factor: Number of Queue

Packet scheduling schemes can also be classified based on the number of levels in the ready queue of a sensor node. These are as follows.

Single Queue

Each sensor node has a single ready queue. All types of data packets enter the ready queue and are scheduled based on different criteria: type, priority, size, etc. Single queue scheduling has a high starvation rate.

Multi-level Queue

Each node has two or more queues. Data packets are placed into the different queues according to their priorities and types. Thus, scheduling has two phases: (i) allocating tasks among different queues, (ii) scheduling packets in each queue. The number of queues at a node depends on the level of the node in the network. For instance, a node at the lowest level or a leaf node has a minimum number of queues whilst a node at the upper levels has more queues to reduce end-to-end data transmission delay and balance network energy consumptions. Figure 4 illustrates the main concept behind multi-level queue scheduling algorithms.

The work done by Lee E.M et al. [63] proposes a multi-level-queue scheduler scheme which uses different number of queues according to the location of sensor nodes in the network. This approach uses two kinds of scheduling: simple priority-based and multi-FIFO queue-based. In the former, data enters the ready queue according to priority but this scheduling has a high starvation rate. Multi-FIFO queue is divided into maximum three multi queues, depending on the location of node in network. If the lowest level is L_k, nodes that are located at level L_{k-1} have only one queue but there are two queues for nodes at level L_{k-2}. Each queue has its priority set to high, mid, or low. When a node receives a packet, the node decides the packet's priority according to the hop count of the packet and accordingly sends it to the relevant queue. The work done by Karimi E. and Akbari B. [55] also proposes a priority queue scheduling algorithm for WMSN. In this scheduling scheme, buffer space of intermediate nodes is divided into four queues to hold three different types of video frames and one regular data frames. Data in the first three queues have the highest priority and are scheduled in round-robin fashion. Data in the fourth queue is transmitted when the first three queues are empty. However, these scheduling schemes do not consider variable number of queues based on the position of sensor nodes to reduce the overall end-to-end delay.

2.2.5 Discussion

In literature, sensor nodes mostly use First-Come-First-Served (FCFS) [96] and Multi-level Queue scheduling schemes. In FCFS scheme, data traffics are processed in the order of their arrival times. Other priority-based approaches provide priority to data packets based on the number of hops they travel towards base station (BS). These scheduling schemes do not consider providing the highest priority to real-time data and incur a large end-to-end data transmission delay for real-time data. Thus, these approaches are not suitable for Wireless Sensor Networks (WSNs) that are used for emergency and real-time monitoring applications.

2.3 Routing Protocols

Routing is a challenging issue in ad hoc networks in general and in Wireless Sensor Networks (WSNs) specifically due to their unpredictable topology and lack of infrastructure. This raises the need for routing protocols that are highly dynamic and adaptive to topology changes. And these protocols should also provide alternatives to any failed route in order to maintain connectivity of the whole network. Routing protocols differs from one application to another; no single protocol was found to suit all diverse WSNs applications' requirements. In general, since sensor nodes have limited bandwidth, energy, storage capacity, and processing speed designing a fault tolerant and energy efficient routing protocol that tradeoff between the energy and scalability is an important research issue in WSN.

Routing protocols of WSN can be classified as flat, clustering, and location-based [79, 26, 26, 80, 7, 40, 107]. In the following subsections, we present existing routing protocols of WSNs with their limitations and comparison.

2.3.1 Flat Routing Protocols

In flat routing protocols, sensor nodes are homogeneous in terms of the initial energy and role. To perform the same task, nodes coordinate with each other. Base Station (BS) transmits messages to specific regions of the network and waits for data to be received from the sensors of those regions. These protocols are scalable but not energy efficient. Directed Diffusion (DD) [42, 18, 46], Flooding [2, 18], Sensor Protocol for Information Negotiation (SPIN) [2], Gossiping [2] are the names of flat routing protocols.

2.3.1.1 Flooding

In Flooding routing protocol, each sensor node broadcasts its sensed data to neighboring nodes that increases network traffic and energy consumptions. Moreover, the received data by neighboring node is again broadcasted and thus, received by the originator node. This problem is known as implosion. For example, in Figure 2.6, each of the nodes A, B, C and D are sending data to each other. Moreover, in flooding, sensors cover overlapped geographical areas and so may receive overlapped pieces of information. Resource blindness is another limitation of flooding. Sensors do not change the services they provide and activities that they do based on their resource constraints, so they are resource blind.

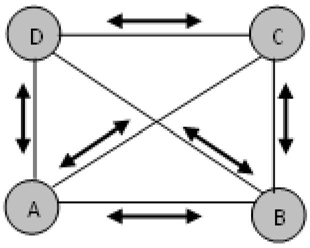

Figure 2.6: Flooding

2.3.1.2 Sensor Protocols for Information via Negotiation (SPIN)

Sensor Protocols for Information via Negotiation (SPIN) is designed to overcome implosion, overlapping, and resource blindness problems of the flooding routing protocols. In SPIN, whenever a node has data to send, it does not flood the data to all its neighbors as in flooding routing. It first advertises its data to all neighbors by sending some descriptive information about what data it has. Then any neighbor node, which does not possess this information and is interested in it, will send REQ packet requesting this data. Afterwards the advertiser will send its data only to the nodes that show interest in that data. Figure 2.7 illustrates the operation of SPIN protocol.

In SPIN [1], sensor nodes only need to know about its neighbors and thus, consumes little energy in the computation. However, SPIN does not use clustering of the nodes. It still has traffic overhead due to advertising and requesting data through the network and long latency due to waiting for advertise-request negotiation. Thus, this protocol is not scalable.

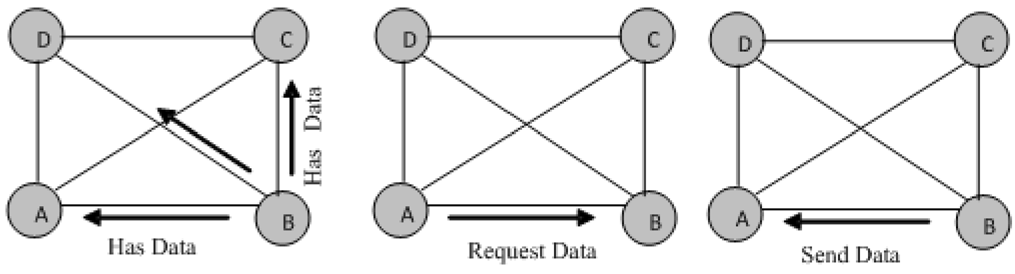

Figure 2.7: Sensor Protocols for Information via Negotiation (SPIN)

2.3.1.3 Directed Diffusion (DD)

Directed Diffusion (DD) [42] is a data-centric routing protocol, in which observers request the data based on criterions like sensors geographical location, sensed readings exceeding certain thresholds, etc. It employs a request-response mode in its operation.

The protocol is based on four main elements:

1. The interest: It is a query that the observer sends to the network requesting a certain piece of information. The interest is in the form of data pairs describing the request details, such as {area = A_1, temperature = +20 degree Celsius}.

2. The gradient: It specifies the rate at which the source should send the information to the sink, together with the direction of how to reach the interested observer.

3. Data messages: are the data being transferred from source to destination.

4. Reinforcements: an interest packet is resent through the network to select a certain number of data paths from multiple paths.

DD is an application-aware protocol [1], which routes data based on the application needs. It can also save energy by selecting paths requiring the lowest energy cost. The interested (source) node broadcasts its interest message to all neighboring nodes at a lower data rate. Neighboring nodes forward it in turn to their neighbors until all nodes in the network receive this interest message. When a targeted node senses data related to the interest message it will send the information through the network. Intermediate nodes may cache and aggregate these pieces of information in their way to the destination. When the initial data arrives at the sink, it will send a reinforcement interest packet at a higher data rate. This packet determines a specific path for data delivery between the sink and source. As the path is determined and the data rate is increased, the data delivery process will be faster. Also this protocol can support the event-driven scenario. In this scenario the source mode will broadcast their data to all nodes. After any interested sink receives the data it will reinforce specific path to that source and increase data rate in a way similar to the request-reply scenario.

DD is better than flooding and multicast in terms of energy dissipation. This is because the data rate is only increased after the interest happens. And as the protocol is application-aware, data is not transmitted over all paths, like in flooding. Only one path, the best energy-wise, is chosen and reinforced from a bunch of paths for data dissemination after the interest happens. Moreover, there is no data duplication, which is better than flooding protocols. However, in DD, the network lifetime is reduced due to the periodic broadcast of the interests and the low data rate paths. Data aggregation requires time synchronization which is a costly task in WSNs.

2.3.1.4 Limitations

In general, flat routing protocols suffer from imbalanced energy consumptions of nodes, poor scalability (i.e., if the number of nodes increases in the network the average number of hops between the source and BS also increases) and poor robustness for not being able to send data to BS due to the failure of a node [98, 99].

2.3.2 Location-Based Routing Protocols

In these protocols, locations of sensors are addressed using Global Positioning System (GPS), Time Difference of Arrival (TDoA), Time of Arrival (ToA) or other techniques. Sensors are densely deployed. Hence, these protocols are expensive. Geographic Adaptive Fidelity Protocol (GAF) [80] is a location based routing protocol.

2.3.3 Hierarchical or Clustering Routing Protocols

To overcome the limitations of flat routing protocols, hierarchical routing protocols are used that are also known as clustering protocols. In clustering protocols of WSN, several sensor nodes in the communication range of each other form a cluster. Each cluster has a Cluster Head (CH), which coordinates all non-CH or member nodes of the cluster. There may be a number of BS that is also known as a sink in a WSN to communicate to other networks. CH aggregates data received from all member nodes of a cluster and sends to BS. Besides CH, there exist some sensor nodes in a cluster known as gateways which are used for inter-cluster communications. Clustering protocols are more useful in the context of energy efficiency because it produces limited useful information from large amount of raw sensed data and hence, transmitting this precise information to BS of the network consumes less energy. Low Energy Adaptive Clustering Hierarchy (LEACH) [38], Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [70], Threshold Sensitive Energy Efficient Sensor Networks (TEEN) [77], Adaptive TEEN (APTEEN) [78], Low Energy Static Clustering Scheme (LESC) [40], and Optimal Energy Aware Clusering [31] are examples of clustering protocols. Clustering protocols can be classified as (i) static and (ii) dynamic. In static clustering protocols, CHs are fixed and deplete energy faster than other sensor nodes and fail which is known as hotspot problem. Dynamic clustering protocols solves hotspot problem by periodically changing CH in a cluster. Block diagram of clustering protocols is shown in Figure 2.8. We present some existing clustering protocols of WSN in the following subsections.

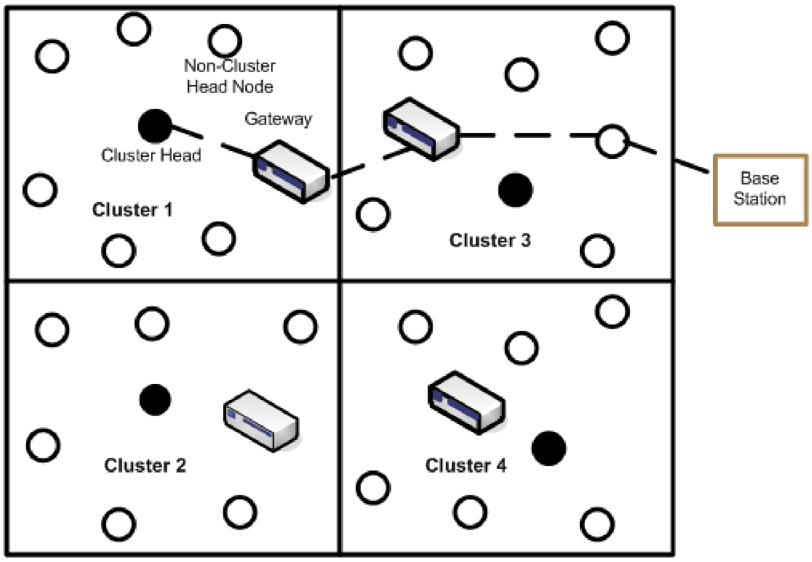

Figure 2.8: Clustering protocols of Wireless Sensor Network

2.3.3.1 Low Energy Adaptive Clustering Hierarchy Protocol and its Variants

Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [38, 26] works well for homogeneous networks, where every node has the same initial energy. LEACH works in rounds, where a round comprises cluster formation and steady phases. In the cluster formation phase, a cluster is formed and $p \times n$ sensor nodes are selected as cluster heads (CHs) for the proper utilization of energy where, n is the number of sensor nodes, and p is the desired percentage of CH. Otherwise, if only one node is selected as CH it will fail due to the shortage of energy. The decision to select a node as CH is based on a threshold value $T(n)$. Each node A chooses a random number, R_n (between 0 and 1). If $R_n < T(n)$, A is selected as a CH in the current round, where,

$$T(n) = \frac{p}{1 - p \times (r \mod \frac{1}{p})} if n \in G \tag{2.1}$$

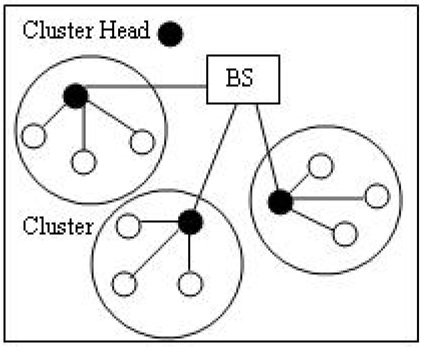

Figure 2.9: LEACH routing protocol

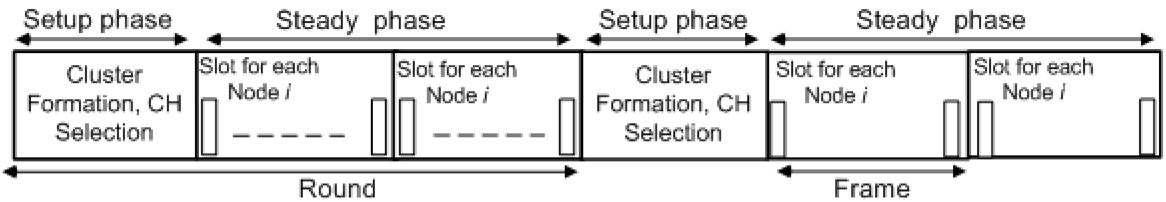

Figure 2.10: Timeline of different phases of LEACH routing protocol

Moreover, the suggested percentage of CH of the networks and the number of time a node is selected as a CH are also decision criteria for a node to be a CH in the current round. The steady state is divided into many frames and the CH assigns time slots for each non-CH node using TDMA scheme in each frame. Each node sends data to the CH at its allocated timeslot. At the end of each round, the CH collects and aggregates data and sends to the BS. Figure 2.10 illustrates the time line of setup and steady phases.

However, CHs communicate directly with the BS in LEACH protocol, which consumes much energy of CH if BS is far away from the CHs (Figure 2.9). A new cluster formation is initiated in every round, which require a large number of data transmission and is not energy efficient. Moreover, occasionally all CHs exist in a close area (as CH rotates in a cluster). Then, non-CH nodes require more energy to communicate with CHs.

In LEACH, security is obtained by providing CH authentications, data confidentially, etc. However, LEACH does not provide two party data authentications. Thus, a secured LEACH protocol, namely sec-LEACH [12] is proposed that uses random key distributions scheme. This key distribution approach is designed only for a node to communicate with a static set of neighboring nodes. In this approach, a key pool P of size S is generated. A key ring with m random keys selected from P along with a unique ID, ID_A, is assigned to a node A. ID_A is used to seed a random number generator to generate m random numbers R_A, which represent the key IDs assigned to the node ID_A. Each node also has a key to share with the BS. In the advertisement phase, CHs broadcast their IDs so that each node can use these IDs to produce a set of m keys of a CH using pseudo random number generator. A node joins in a cluster if a key of that node matches with a key of the CH of that cluster. This condition might create orphan nodes (i.e., nodes reside outside of a cluster) by restricting a node to join in the closest cluster. Hence, orphan nodes send data directly to a BS, which can be a far away. Thus, sec-LEACH consumes more energy. Moreover, nodes in a cluster might be a far away from the CH. Hence, this key distribution method is not energy efficient. Thus, a secure grid based LEACH protocol (GS-LEACH) [12] is proposed, which tradeoffs between security and energy efficiency by providing random key distributions and a controlled deployment of nodes in a grid.

In GS-LEACH protocol, n sensor nodes are placed around each grid point in a region R that form a cluster and selects few CHs to communicate with the BS. A BS resides outside or far away of R and is assumed to have more energy and processing capabilities. CH rotates in the cluster for a proper distribution of energy and performs data fusion. Before a node is deployed in a cluster, it is assigned to a randomly generated m keys from a key pool P of size S for communicating with other nodes in the cluster. In addition to these keys, each node has a key to communicate with BS. During the setup phase, each node x elects itself as a CH and sends its key information to other cluster members. The nodes respond to x by sending a Join_Request message and common keys ID, if they have common keys with x. In the steady state, CH uses TDMA scheme and sends timeslot information to each node of the group (cluster). Every node is allowed to sense, process and send encrypted data to CH only in its allocated timeslot and thus, energy is saved. After getting encrypted data from each node, CH decrypts it using the common key. In GS-LEACH, each node is allowed to transmit data in a small region around the grid point, which ensures less communication overhead. In sec-LEACH a node is allowed to communicate with any CH (based on the common keys) even if the CH is far away from the node. Thus, sec-LEACH is not energy efficient.

2.3.3.2 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [70] is a chain-based protocol built on the LEACH paradigm. In every round, each node finds its closest neighbor using greedy algorithm and construct a chain. A node is randomly selected as a leader node that sends a control token to the nodes in chain to start data transmission. Initially, the node at the end of the chain sends its sensed data to its neighbor. The neighbor receives the data and fuses it to its data to form a packet of the same length. Then it sends the new packet to its neighbor and so until the data arrives at the leader node. The leader adds its own data and then sends the data summary of the whole chain to the base station using direct communication.

In PEGASIS, nodes only send to their neighbors. So the transmission load and consequently the power needed is lower. The leader receives only a single summary packet for all sensed data. So it reduces the overhead caused by the cluster head receiving from all nodes in LEACH. However, PEGASIS assumes the availability of location information for the other nodes. Though this protocol can still work without this information but the nodes will expend some extra energy to find their closest neighbor. Moreover, the chaining causes a long latency till the data arrives at the destination. And the quality of sensed data may not be good due to successive fusing.

2.3.3.3 Threshold Sensitive Energy Efficient Sensor Networks Protocol

Threshold Sensitive Energy Efficient Sensor Networks (TEEN) Protocol [77, 80] is a hybrid of data centric and clustering protocol that reduces the number of data transmissions to BS by implementing soft and hard threshold. Hence, the lifetime of a network increases. Initially, clusters are formed and CHs are selected by BS. CHs broadcast these two thresholds to all non-CH nodes. Hard threshold is the minimum sensed value of a subscribed event by BS. When a node senses an event (e.g., temperature) with hard threshold, the node turns on its radio to transmit data to the BS. Once hard threshold is sensed, a node transmits data only if the event value is changed by the soft threshold. However, in TEEN, BS is responsible for cluster formation and CH selection which creates a lot of network traffic and consumes more energy. Moreover, TEEN does not work for periodic transmission of sensed data.

To overcome the limitation of responsive data collection, only when threshold value is reached, in TEEN, Adaptive TEEN (APTEEN) is proposed [78, 79], which works both in periodic and responsive data collection mode. APTEEN has also the drawback of engaging the BS in creating clusters and selecting CHs. Moreover, clusters are formed here in multiple levels and require more communications. Thus, it is not energy efficient. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [70] protocol overcomes the drawback of having multiple levels of clusters in APTEEN by creating a chain of nodes.

2.3.3.4 Periodic, Event Driven and Query Based Routing Protocol

Most protocols of WSN are not designed as fault tolerant. The work done by A. Martirosyan and A. Boukerche [79] proposed Periodic, Event Driven and Query Based Routing (PEQ) Protocol that is a reliable, energy efficient, low latency and fault tolerant clustering protocol. In PEQ, the network is organized as a hop tree, where nodes at different levels are determined by value of the hop count. PEQ uses subscription/publish paradigm for periodic event notification. BS subscribes to the network for some specific events to notify. Whenever a node detects the subscribed events it generates an event notification packet that contains forwarding and destination addresses for that event. Forwarding addresses are the addresses through which the BS transmitted the event subscription packets. The event notification packet will reach BS through all forwarding addresses. PEQ uses Acknowledgment-based fault tolerance mechanism. Whenever a node x sends data to its neighbor y but does not receive acknowledgement from y within a predefined period of time, x assumes that y has failed. Then x broadcasts a packet to its neighboring nodes to search for a new forwarding node [80]. Neighboring nodes reply to x with a packet that contains their ID and hop count value. The neighbor with the minimum hop count is elected as a new destination. However, PEQ is not considered so much energy efficient since it is not cluster-based and use tree structure. Hence, Cluster-based PEQ (CPEQ) [80] is designed, where nodes with more remaining energy are selected as CHs and clusters are formed. When nodes in a cluster sense any event it sends the event's value to CH. If a CH receives any redundant data it performs data aggregation, fusion and filtering. Hence, BS will receive non-redundant data from CH. CPEQ uses the same hop tree configuration method as of PEQ with the similar CH selection mechanism as of LEACH. If percentage of CH nodes is 3%, the probability for each node to become CH will be 0.03 ($CH\_Probability$). A random number R_n (between 0 and 1) is generated by a node x and if $R_x < CH_{Probability}$, x works as an elector. Then, x initiates CH selection by sending energy request packet to all neighboring nodes. Neighboring nodes reply by sending a packet to CH, which contains node's address and residual energy. Elector will select nodes with more residual energy as CH and notifies all other nodes. CH selection is carried out periodically in this protocol.

2.3.3.5 Low Energy Static Clustering Scheme

In static clustering protocols, CHs deplete energy faster than other sensor nodes and fail. This is known as a hotspot problem. In [40], Huang B. et al. propose a Low Energy Static Clustering Scheme (LESCS) for WSN, which solves the hotspot problem of static clustering by distributing the energy consumption equally over all nodes. Hence, this scheme increases the network lifetime. LESCS considers all sensor nodes homogeneous, energy constraint and fixed. BS is fixed and placed at the outside of WSN and the whole network fails due to the failure of a certain percentage of sensor nodes.

This protocol has (i) Centralist Network Clustering Calculation (ii) Cluster formation and (iii) Intra-Cluster Scheduling phases. In Centralist Network Clustering phase, BS divides the whole network into a few numbers of clusters and also selects CHs for each cluster. Once clusters are formed and CHs are selected they are fixed and the BS broadcasts this information to whole network. In the cluster formation phase, each non-CH node joins in a cluster whose central point is closest to that node. The CH records the position and energy information of each node in its routing table and also assigns ranking/sequence number $0, 1, 2, \ldots k-1$ to the sensor nodes. In Intra-Cluster Scheduling phase, CH will assign a gateway (a sensor node) for its cluster to communicate with neighboring clusters. The gateway of one cluster sends information to the gateway of neighboring cluster for inter-cluster communication. Gateway will change in each round. Otherwise, the energy of the gateway will deplete very fast and eventually, it will fail.

A gateway performs data collection, fusion, and transmissions to the gateway of neighboring clusters using multi-hop data transmission in LESCS. If a sensor permanently works as a gateway it depletes energy very quickly and fails (hotspot problem). To avoid this problem, the most residual energy node is selected as a gateway in the next round. Initially, CH predicts the residual energy of each node, assigns the most residual energy node as a gateway and broadcasts this information to the network. If any other node x in the network has more residual energy than the assigned gateway, CH reassigns x as a gateway.

2.3.3.6 Dynamic Static Clustering Protocol

The work done by F. Bajaber and I. Awan [8] proposed a clustering protocol, namely Dynamic Static Clustering (DSC) for Wireless Sensor Networks (WSN). DSC protocol has dynamic and static cases. In the setup phase of dynamic case the BS forms clusters and selects CHs for each cluster based on the residual energy and positions of sensor nodes. Then, BS broadcasts CH ID to all nodes. A sensor node will be a CH if its ID matches with the CH ID. In the steady phase, CH uses TDMA scheme by dividing each frame into x number of timeslots, where x is the total number of non-CH nodes in that cluster. A non-CH node transmits data to CH only in its allocated timeslots and saves energy by turning its radio off in all other time slots. When a round is completed, data transmitted by all non-CH nodes are aggregated and sent by the CHs to the BS. In the next round, the CH of a cluster selects a node as CH, which has the most remaining energy and informs the BS. Figure 2.11 represents the working principle of DSC protocol.

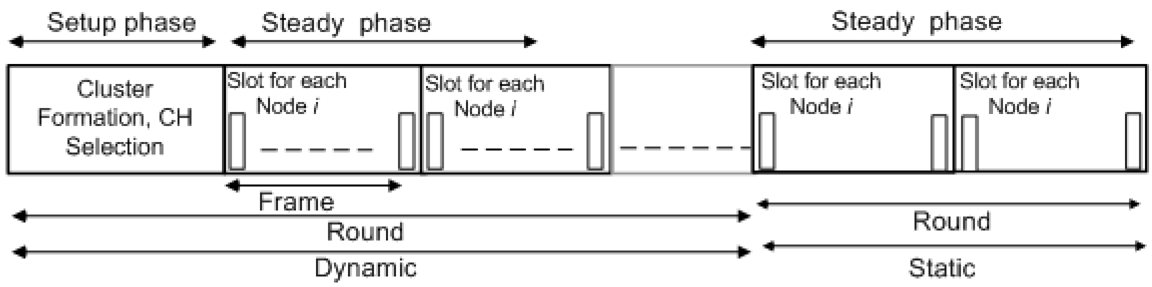

Figure 2.11: Dynamic Static Clustering (DSC) protocol

Static case has only the steady phase, which is similar to that of dynamic case except for after a certain number of rounds a new cluster setup phase is initiated. Static case has less transmission overhead but CH depletes energy faster than the dynamic case. Thus, DSC balances energy consumptions by properly distributing its work into dynamic and static case.

2.3.3.7 Other Protocols

Since CH directly sends data to BS energy consumption is more in LEACH. To overcome this limitation of LEACH, Zhenhua Yu et al. [30] proposed Energy Aware Cluster Routing (ECR) protocol where, a node with the maximum residual energy is selected as CH and CH sends data to BS through other CHs in multi-hop routing. In ECR, BS divides the network into some areas (clusters) based on the location information sent by each node using GPS or other location identification mechanisms as well as selects a node in each area as an auxiliary CH. Then BS sends CH ID and location information to all other nodes and nodes reply to the auxiliary CH, which stores energy and location information sent by each node of an area. Auxiliary CH sends its residual energy to all members of the cluster as well. If energy of a member node X is greater than the energy of the auxiliary CH then X is selected as CH and its location and energy information is sent to all other members. ECR optimizes the number of unnecessary nodes to minimize energy consumptions. However, cluster formation or CH selection is done several times in ECR and hence, is not considered as energy efficient.

However, most of the routing protocols are designed for homogeneous sensor networks. The work done by C. Duan and H. Fan [24] proposed a Distributed Energy Balance Clustering (DEBC) Protocol for heterogeneous WSN, where CHs are selected based on the ratio of remaining energy of a node and total network energy. The probability of a high energy node to be a CH is much higher than that of low energy node and thus, DEBC provides better performance in terms of energy consumptions and network lifetime. However, in most clustering protocols, all sensors take part in CH election and the energy of sensors is dissipated quickly if the CH election is performed frequently. Hence, Zhi-feng Duan et al. [25] proposed Shortest Path (SP) routing protocol for Multilayer Mobile Wireless Sensor Networks (M2WSN) that classifies nodes as S node (normal sensors), Fuse Node (F), and Control Node (C). S nodes have only limited storage and processing capacity and are responsible for collecting data from the environment and transmitting data to the nearest (one hop) F nodes, which maintain routing tables, receive, aggregate and send data to C nodes through other F nodes in multi-hop. F nodes have GPS and so know their position. C nodes control the whole network and work as a gateway between Sensor Network and Internet. Since S nodes do not take part in such kind of CH or F node election process and remain in sleep mode most of the time, SP protocol reduces energy consumptions. However, most clustering protocols of WSN do not consider position information of sensors, which is very important to take appropriate measures for emergency events. In [80] authors present Geographic Adaptive Fidelity Protocol (GAF). GAF is a location-based routing protocol designed especially for ad hoc networks but also works for WSN. Some nodes remain in sleeping mode to preserve energy without affecting the network connectivity. GAF forms virtual grid (geographic area) for a set of nodes, which have the same communication ability with the nodes of neighboring grid. This protocol also represents hierarchical clustering where a grid or geographic area corresponds to a cluster. However, this protocol is expensive since nodes use GPS for locating position.

Among other clustering protocols, Optimal Energy Aware Clustering [31] performs theoretical analysis to introduce balanced k-clustering approach that balances energy consumptions among master (or cluster head) and member nodes of the cluster by reducing distance between master and member nodes and this approach. This approach is modeled as min-cost flow problem [31] and can be considered to achieve optimal clustering topology. Energy Efficient Protocol with Static Clustering (EEPSC) partitions the network into some static clusters and uses residual energy of sensor nodes to select CH. However, if CHs reside at the boundary of clusters energy consumptions for intra-cluster communications increase. Thus, the work done by Chaurasiya S.K et al. [19] proposes Enhanced Efficient Protocol with Static Clustering (E^3PSC), which also considers the spatial distribution of sensor nodes along with residual energy to select CH. Adaptive Clustering Algorithm based on Energy Restriction (ACAER) [39] is another clustering protocol that selects CH based on residual energy and network coverage rate (i.e., the quality of communication). Thus, this protocol is important for WSN having sensor nodes with different communication range. Cluster formation for static clustering protocols are done at the beginning of a predefined round even if remaining energy of CH is at critical stage. In [28], Elhawary M. and Hass Z.J. propose a co-operative clustering protocol, where each node on the data transmission path from the source to destination node becomes CHs. Each CH recruits neighboring nodes to coordinate its transmission towards destination and thus, forms a cluster to reduce packet delivery failure. In [90], authors propose Hybrid Clustering Approach (HCA), where cluster formation is done on-demand at the beginning of the next round if the remaining energy of CH is lower than a threshold value. However, in these protocols, nodes do not co-operatively to send data to BS. However, HCA and most existing clustering protocols do not provide fault tolerance. LEACH-SM [9], FZ-LEACH [56], Pendulum [3], AWARE [105], EEDC [67], EER [11], and TSEP [106] are the name of some recent and popular clustering protocols of WSN.

2.3.4 Discussion

From the literature, we find that some of the routing protocols use TDMA schemes to allow nodes to send data to Cluster Head (CH) in round robin fashion in its allocated timeslot. These schemes are not considered energy efficient because sensors might not sense an event of interest or data value at every timeslot and sending data at every timeslot consumes much energy. Moreover, these techniques require CH selection or cluster reconstruction at each round or after a certain number of rounds.

Table 2.3: Comparison of clustering protocols of Wireless Sensor Network

Features	LEACH	TEEN	APTEEN	DSC	CPEQ	LESCS
Single-hop (CH to BS)	√	X	X	√	X	X
Multi-hop	X	√	√	X	√	√
Cluster is formed by BS (increases network traffic)	X	√	√	√	X	√
CH sends aggregated data to BS	√	√	√	√	X	X
CHs relay events notification to BS (requires long transmission)	√	√	√	√	X	X
Periodic,event,query-based notification	√	X	√	X	√	X
Provides fault tolerance	X	X	X	X	√	X

Most of these protocols are not fault tolerant and use tree structure. Thus, failure of a node results in many other nodes to be disconnected from the network. Moreover, failure of a single node or CH results in the whole network or clusters to be reconstructed, which requires a large number of message transmissions among nodes and dissipates much energy of the network. Table 2.3 presents the comparison of several cluster-based routing protocols of WSN on some features.

2.4 Data Aggregation Approaches

Data aggregation [18, 41, 42, 45, 46, 61, 117] reduces the number of data transmissions and network energy consumptions by aggregating redundant data packets in intermediate nodes especially, in dense sensor deployment, where several nodes sense the same event. Aggregating a large number of non-redundant packets also represent better data accuracy. For instance, temperature data from 20 sensors has better accuracy than that received from 2 sensors. However, a parent node waits a long period of time for aggregating data received from all child nodes that result in longer end-to-end data transmission delay. Thus, designing an efficient data aggregation protocol that trades-off between network energy consumptions, data accuracy, and end-to-end delay is significantly important.

Data aggregation process in sensor networks consists of several steps [18].

Step 1: A query for the required data is distributed through the network to find the node that senses the relevant data.

Step 2: Each sensor selects a node as its parent from which it receives the query announcement. The resulting topology is known as the aggregation tree.

Step 3: Non-leaf nodes of the tree aggregate data received from their children and transmit the aggregated data to their parent nodes.

Step 4: Once the aggregation tree is created, nodes schedule the period of data transmission and reception.

Step 5: Parent nodes wait a specific time period for data to be received from their children nodes and then transmit the partially aggregated data at the upper level of the tree.

The following subsections present several existing data aggregation methods with their limitations, classifications, and comparisons.

2.4.1 Existing Approaches

Scalable and Unified Management And Control (SUMAC) [46] is a large scale Wireless Sensor Network (WSN) architecture that uses a medium range mesh network as a bridge between geographically dispersed sensor clusters and Internet. SUMAC provides users the full data ownership and transmission of data within their own network, which is inexpensive, fast, secured and energy efficient. In SUMAC, a high level setting (by users) triggers a background process to set a default data aggregation level and also to determine rules and conditions to modify the default aggregation level. For instance, default aggregation level can be set to average most sensors data in a single cluster (to save energy). When an event of interest occurs sensors can automatically slide down the aggregation level to enable node to send raw sensor data. SUMAC aggregation contains energy cost for a path generation. Nodes share their energy consumptions, delay and buffer size with the immediate neighbors. This method is coupled with a visual interface to control the aggregation (static or dynamic) based on the assigned aggregation rules such as hop count, resolution, sensors value, sensor tags, and GPS coordinate. In static aggregation, a user selects sensor nodes based on GPS coordinate, label or ID and sets the selected node to aggregate or to forward their data. Server statically sets the node aggregation level and nodes follow the rules until a new rule arrives from the server. In dynamic aggregation, aggregation level changes if an event of interest occurs. For instance, if the current aggregation level is 2 and if any node at level 3 detects an event of interest aggregation level slides down to level 4 so that data at level 3 are not aggregated. Users set the maximum or minimum threshold for sensed values such as temperature, light. If the temperature exceeds the threshold the sensor nodes automatically stop aggregating the packets and also instruct neighboring nodes to do so (to monitor the region around this event of interest which is also known region-based aggregation). Users can enable or disable region-based aggregation. If it is disabled the sensor node that detects the event will override aggregation while neighboring nodes follow the normal aggregation policy.

In Directed Diffusion (DD) [42, 41], interest messages flow from the sink to the source using expensive flooding (Interest propagation). Then data messages flow from the source to the sink initially along multiple paths towards the sink (data propagation). Later, the sink selects only a number of paths based on the quality of data received. Thus, the total number of nodes in data transmissions is reduced. To alleviate expensive flooding for interest propagation in DD clustering approaches are used, where interest messages are only sent to cluster head (CH) and gateways. Chatterjea S. and Havinga P. [18] propose Clustered Diffusion with Dynamic Data Aggregation (CLUDDA) that improves energy and network efficiency by integrating clustering into Directed Diffusion (DD) and allowing nodes to collect and aggregate data by including entire query definition with interest message. CLUDDA is also divided into two main phases, interest propagation, and data propagation. The format of interest packet is significant in interest transformation, dynamic aggregation point formation that also allows nodes to deal with unfamiliar queries. Since interest packets contain the query definition they allow nodes to break down a query into its fundamental components, gather and process data for individual components that results in data reduction. Then nodes transmit these aggregated data to the sink. The work done by Minming T. et al. [84] proposed a layered (clustered) data aggregation model for underground coal mine monitoring based on DD algorithm to overcome its deficiency by using geographical information of sensors' data of different grid clusters. All nodes will be in grid clusters based on their geographic locations. CHs form upper levels and all other nodes form lower levels. Nodes at upper levels collect and aggregate data and send to the central server through the gateway nodes and nodes at lower levels monitor the environment. Hence, the flooding and redundancy of DD is reduced that increases energy efficiency of the network. This method works well for periodic enquiries, real-time alerts (e.g., gas outburst) and continuous monitoring.

The work done by Ying Liang, and Hongwei Gao [68] proposed an Optimal Clustering Algorithm Based on Target Recognition (OCABTR) that collects data periodically and reduces transmission overload and energy consumptions of sensor nodes. When clusters are formed, sensors reside in different clusters might represent the same geographical area in terms of events to sense that increases data redundancy. It is also difficult to aggregate similar data in different clusters. Hence, OCABTR uses genetic algorithm to partition nearby or adjacent nodes (to form cluster) that sense similar events into a cluster. This reduces redundant data transmissions and network energy consumptions. Data aggregation based on dynamic routing (DABDR) [123] is another cluster-based aggregation approach. DABDR creates a tree structure where parents wait a certain period of time for child data. Data packets have a depth field that ensures the direction of data flowing from a sampling node to the sink and a queue length field that makes data packets flow to the nodes that have a long data aggregation (DA) queue. Thus, data packets are concentrated more to make the aggregation more energy efficient.

Tiny Aggregation Approach (TAG) [6] is also a dynamic data aggregation method where, each epoch (time duration) is divided into a number of timeslots. Different levels of tree are associated with different timeslots and nodes of each level of a tree can only send data in their specific timeslots. Hence, synchronization is achieved for sending and receiving data that reduces energy consumptions. However, in this approach, if a node does not receive data for a child at its specified timeslot the unused information of the whole sub-tree rooted at that child will result in an ultimate inaccurate data. Bidirectional Data Aggregation (BDA) [6] adds a label to each query in addition to the basic working principle of TAG. Hence, data transmissions for similar queries can be ignored that reduce energy consumptions. TAG and BDA do not have any fault tolerance mechanism to prevent from fault and inaccurate results. Hence, the work done by Anisi M.H. et al. [6] proposed Fault-tolerant Energy-Efficient Data Aggregation (FEDA) where, if a packet is lost due to a link error or failure between two sensor nodes, other nodes that have overheard the lost packet and have not yet sent their data can aggregate the lost packet with their data packets and send. Moreover, FEDA uses tree structure. During the tree building phase, each node x chooses a node as its primary parent from which x receives a tree building message for the first time and another node as a second or backup parent (for fault tolerance) from which x receives tree building message for the second time. Each packet has a Fault Status (FS) field. If a primary parent receives a packet with $FS = 0$ from its child it aggregates the data of its child to its own local data and sends towards the destination. Otherwise, the parent only sends its own local data toward the destination. If a backup parent receives a packet with $FS = 1$ from its child it aggregates the data of its child to its own local data and forwards; otherwise, it only sends its local data without any aggregation towards the destination.

Most existing cluster-based aggregation approaches are not adaptive, i.e., are not suitable to work in environmental changes and are restricted to the types of sensing scenarios. In [110] Woo-Sung J. et al. proposed an adaptive data aggregation technique that can select any suitable clustering method based on the application requirements and thus, reduces network energy consumptions, and increases data transmission reliability. As neither static nor dynamic aggregation is suitable for target tracking operations, the proposed scheme selects the static clustering-based aggregation when the data traffic is high, and adaptively switches to the dynamic clustering aggregation when the data traffic is low. Data traffic threshold is calculated based on the number of data packets the sink receives over a certain period of time, which is also used to switch between each data aggregation method [110].

Although data aggregation is important to reduce redundancy it increases end-to-end delay because aggregator nodes wait for the arrival of data from a large number of sensor nodes. Thus, the work done by Mirian F. and Sabaei M. [85] propose a layered data aggregation to meet the application requirements for end-to-end delay, accuracy and energy consumptions. The number of clusters in each layer is selected so as to control the accuracy and delay to deliver the aggregated data from sources to sink within a specified time period. In [85] a delay sensitive feedback control data aggregation (DSFC) model is presented that solves the trade-off issue between the aggregation waiting delay and aggregation accuracy.

Among other recent data aggregation approaches, the work done by Meng L. et al. [82] proposes, DACP, an energy efficient data aggregation approach based on clustering and prediction. Initially, the sink node forms clusters and cluster head (CH)s. Sink receives data from CH and generates predicted data based on all historical recorded data. In the prediction phase, sink broadcasts predicted data. Member nodes of a cluster compare the predicted data with the sensed data and decide whether to transmit the sensed data or node. Thus, number of data transmission is reduced and network lifetime is prolonged. Another clustering and prediction-based data aggregation approach is proposed in [44], which works by adaptively enabling/disabling prediction schemes for clustering-based continuous data collection in WSN. In [91], another cluster-based data aggregation approach of WSN for the structural health monitoring is proposed. This approach distributes tasks into different types of nodes in a cluster, and integrates filtering roles based on node's type. This approach also integrates a role-based data aggregation middleware, and SQL-like user interface to support flexible task configurations. The work done by Xu J. et al. [115] investigates the effect of scheduling and aggregation tree construction on the performance of data aggregation in terms of network energy consumptions and delay. For instance, the Connected Dominating Set Tree (CDST) has the best performance in term of network energy consumptions and delay whereas the the Center at Nearest Source (CNS) tree has the worst performance.

2.4.2 Classifications

By reviewing the literature on data aggregation methods of WSN we can classify these methods in several ways, which are presented as follows.

2.4.2.1 Based on the Structure of Networks

Structured data aggregation is used in clustering routing protocols of WSN. Cluster-based data aggregation protocols reduce the amount of transmitted information to the sink and save energy. LEACH [38] is a cluster-based data gathering protocol. The details of LEACH protocol has been presented in Section 2.3.3. Among other structured data aggregation methods, PEGASIS [70, 17] is a chain-based data aggregation scheme where, all nodes are organized into a chain and communicate with the base station by turn. The work done by C. M. Chao and T. Y. Hsiao propose an optimal multicast data aggregation tree [17]. In this topology, a data is transmitted through a path with the least number of hops. Data aggregation tree can also be created using greedy algorithms, approximation algorithms, and distributed approximation algorithms. In structured data aggregation protocols, if sensor nodes are far away from CH, they might dissipate much energy in data communications.

Structure-free data aggregation is normally used for flat routing protocols. Challenges of structure-free data aggregations are finding the next hop in data routing and scheduling the waiting list of nodes (i.e., determining which node will wait for whom). Data Aware Anycast Randomized Waiting ($DAA + RW$) [17] is a structure-free event driven data aggregation scheme. In this approach, each node that has data to transmit sends a Request-To-Transmit (RTS) message to neighboring nodes. The nodes that receive the RTS message are candidates for the next hop on its data transmission paths to base station (BS). However, nodes that have the same data packets to transmit or are closer to the BS respond with Clear-To-Transmit (CTS) messages. In the randomized waiting, each node that has data to transmit waits for a random period of time before starting its transmission. This will reduce the number of data collision. Aggregation efficiency will be high if the nodes close to the sink have a longer waiting time and vice versa. However, $DAA+RW$ scheme has the similar performance as the structured data aggregation approaches and the randomized waiting time mechanism can be further improved.

2.4.2.2 Based on the Traffic Pattern

Data aggregation is classified as static and dynamic based on the traffic pattern. Static aggregation is good for an unchanging traffic pattern and has a low maintenance cost but unstable for event triggered networks due to a long delay and sink stretch. Dynamic aggregation is used for networks with a frequent changed traffic pattern. As the location of the source nodes always change aggregation point is also changed (dynamic) considering that aggregation is always performed very close to the data source. Dynamic aggregation reduces communication cost but has a high maintenance overhead. TAG, FEDA, and BDA [6] are examples of dynamic data aggregation schemes.

Data aggregation is very important for target tracking applications because of the movement of targets involve a large number of sensors that result in redundant data. Static clustering-based data aggregation technique has advantages when there are multiple targets with high velocity. Dynamic clustering-based data aggregation approaches are suitable for a few number of targets with low velocity [110].

2.4.2.3 Based on the Precision of Information

Data aggregation can be classified as lossless and lossy [41]. In lossless aggregation, detailed information in data is preserved by eliminating redundant information. On the contrary, lossy aggregation discards detailed information in data and hence, data quality is degraded. Timestamp and packing aggregations are examples of lossless aggregation. Timestamp aggregation aggregates events that consist of timestamps such for remote surveillance applications. The redundant information such as hour and minute fields in the timestamp may be minimized. In packing aggregation, several non-aggregated messages are packed into one aggregate without compression. It saves per-transmission overhead, such as packet header.

2.4.2.4 Based on the time of Aggregation

Periodic simple aggregation: each node waits a predetermined period of time, aggregates all data and sends a packet containing the aggregated data.

Periodic per-hop aggregation: each node x sends aggregated data as soon as it hears from all of its children. A timeout is used to detect the lost packets of the child node.

Periodic per-hop adjusted: same as the periodic per hop aggregation except the timeout is scheduled based on the position of the nodes in a distributed tree rooted at the sink.

In the periodic per-hop adjusted timer scheme [123], a node waits t time before aggregating and sending a data packet where, $t = \frac{T}{D(v)} - a$, where, T is the delay request by an application, v is the node, which has queue delay and processing delay denoted by a and depth $D(v)$, i.e., hop count to sink.

2.4.2.5 Others Data Aggregation Strategies

Greedy aggregation [41] is a novel approach to construct a greedy incremental tree. Initially, the shortest path is established from a source node to the sink. Then other source nodes are incrementally added to the nearest point on the existing tree. In greedy aggregation, a node locally decides to send/receive data to/from one or more neighbors. Exploratory samples contain the energy required for delivering the sample from the source to its parent node. Then each node on the tree generates an incremental energy cost that is required to deliver the corresponding exploratory sample to the sink node. The neighboring node that requires least energy to deliver data is more preferable.

Data aggregation is also classified as Spatial and Temporal Data Aggregation. Spatial aggregation aggregates data received from multiple sensor nodes in dispersed geographical areas. Temporal data aggregation aggregates data packets that are received from a sensor at different time. In Region-based aggregation [17], users set a threshold value of a physical parameter in regions such as temperature and humidity. If this parameter value of a sensor exceeds the threshold the sensor node automatically stops aggregating the packets and also instructs neighbors to do so to monitor the region around this event of interest.

2.4.3 Discussion

Static time driven monitoring provides users with highly detailed information. For instance, temperature information from each sensor of a field will be highly redundant, which causes the depletion of sensor energy very fast. Hence, dynamic data aggregation methods that can be controlled by end users should be proposed. In structured or cluster-based data aggregation [17], if sensor nodes are far away from cluster head (CH), they consume much energy for transmitting data to CH. The sink node broadcasts a control packet to the whole network while switching aggregation level in hybrid data aggregation approaches. Therefore, too much aggregation switching may degrade the network performance. To reduce the data aggregation switching, instead of switching the data aggregation method every unit of data traffic measurement, we can configure the switch to occur once every two or three units of data traffic measurement [110]. However, too low aggregation switching frequency can result in data overhead due to the inability to effectively adapt to the changes in the data traffic. Thus, it is very important to find out an optimized switching frequency.

Existing aggregation approaches consider a single sink toward which aggregated events are transmitted. Multiple sinks can be deployed and can generate multiple data aggregation tree rooted at each sink that are expected to increase aggregation accuracy and overall network lifetime by distributing communications for a large scale sensor network.

Defining aggregation functions for different types of application is also very important. For instance, a sensor network application requiring the maximum temperature of a region should use the MAX function to find the maximum temperature. Moreover, a large scale Wireless Sensor Network (WSN) should be application independent that works for different types of applications (Figure 3.2). Some of these applications might require more accurate results, and/or lower end-to-end delay. Thus, aggregation approaches for a large scale WSN should trade-off between delay, accuracy and energy efficiency and consider the factors of delay such as the number of sensors that response a query, and the duration of time an aggregating node waits for data from its children. Table 2.4 presents the comparison of several existing data aggregation approaches on some important features.

Table 2.4: Comparison of different data aggregation methods

Features	DD	FEDA	TAG	DABDR	OCABTR	CLUDDA	SUMAC
Flooding Interest Propagation	√	X	X	X	X	X	X
Initially Data propagate to sink through multiple links	√	X	X	X	X	X	X
Query is divided into several fundamental components	X	X	X	X	X	√	X
Form clusters where interest messages are sent to CH	X	X	X	X	√	√	√
Dynamic data aggregation	X	√	√	√	√	√	√
Static data aggregation	√	X	X	X	√	X	√
Fault tolerance	X	√	X	X	X	X	X
Use tree structure	√	√	√	√	X	X	√
Provide Security	X	X	X	X	X	X	X

2.5 Summary

In this chapter, we presented existing approaches for sensors localization, routing protocols, data scheduling and data aggregation approaches. We also identified limitations in the existing approaches, classified and compared them based on different factors and features. In the next chapter, we will present the architecture of Wireless Sensor Network (WSN) management framework with its major components such as sensors localization, routing, data scheduling, and aggregation approaches.

Chapter 3: Proposed Framework

Sensors have very short data transmission range, and cannot transmit data over a distant central server. Thus, Wireless Sensor Networks (WSNs) are used with the IEEE 802.11 Wi-Fi networks and IEEE 802.16 WiMAX networks having a large transmission power. These networks work as intermediary between WSNs and central servers to transmits sensor's data over the distant central servers. Moreover, designing an efficient WSN network management framework comprising sensors localization, data scheduling, routing and data aggregation approaches is very important for this large scale multimodal WSN architecture.

This chapter is organized as follows. Section 3.1 presents the large scale multimodal WSN overlay architecture with Wi-Fi and WiMAX networks. Section 3.2 presents the architecture of WSN plane for which we design and implement a management framework comprising sensor's localization, routing, packet scheduling, and data aggregation approaches and also demonstrates the working flow of each component in the WSN plane. Section 3.3 presents some general assumptions and defines some important terminologies, and performance metrics.

3.1 Multimodal Network Architecture

This research investigates a scalable WSN architecture that provides end users with full control over individual sensor nodes while abstracting out all intermediate network heterogeneity. The objective is to explore several design issues such as localization, scheduling, routing and data aggregation for a scalable and flexible WSN architecture that relies on multi-modal nodes equipped with IEEE 802.15.4 and IEEE 802.11 in order to use a Wi-Fi overlay as a seamless gateway to the Internet through WiMAX or GPRS networks.

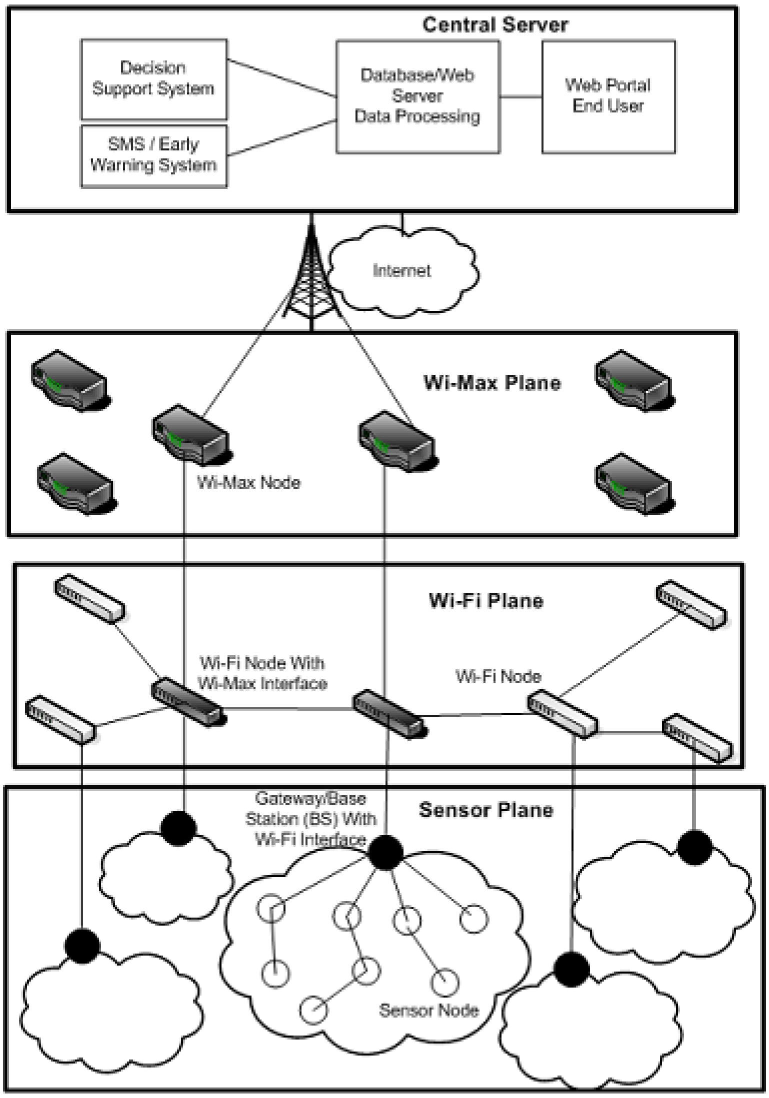

Figure 3.1: Large scale multimodal Wireless Sensor Network architecture

More specifically, the goal is to investigate an architecture that entails three overlay networks: (1) sensor clouds that are deployed around the points of interest, with each sensor cloud forming a locally independent sensor network, (2) a Wi-Fi meshed network, and (3) an infrastructure network plane, such as a WiMAX or GPRS network (Figure 3.2) and design and implement an integrated network management framework comprising sensor's localization, packet scheduling, routing, and data aggregation approaches for the WSN plane of this architecture.

In Figure 3.2, each sensor cloud represents a separate WSN, which is organized as a number of zones or clusters. Sensors of each WSN collect data and send to Wi-Fi network through a gateway node that is a sensor node with Wi-Fi interface. Wi-Fi nodes collect and forward data to WiMAX/GPRS network though Wi-Fi gatewy nodes that also have WiMAX interface. This process continues until data from WiMAX plane are received by the central server that integrates Application, Web, and Database servers. End users can access sensor's data and control sensors through Web interface and server. Database servers store data for future use and also in making decision support systems through data analysis. Early warning systems or application module for early warning is responsible for sending emails, phone, or alarms in case of unwanted abnormal events.

In the following subsection, we present the detailed architecture of WSN plane, and flowchart of the WSN management framework.

3.2 Wireless Sensor Network Management Framework

Figure 3.2 illustrates the architecture of WSN plane, which comprises WSNs for different applications such as agriculture, health, smart home. In this large scale WSN architecture, sensors localization, packet scheduling, routing protocol, and data aggregation techniques are considered as the network management components or tools since they are used to control the operations, and performance of WSN.

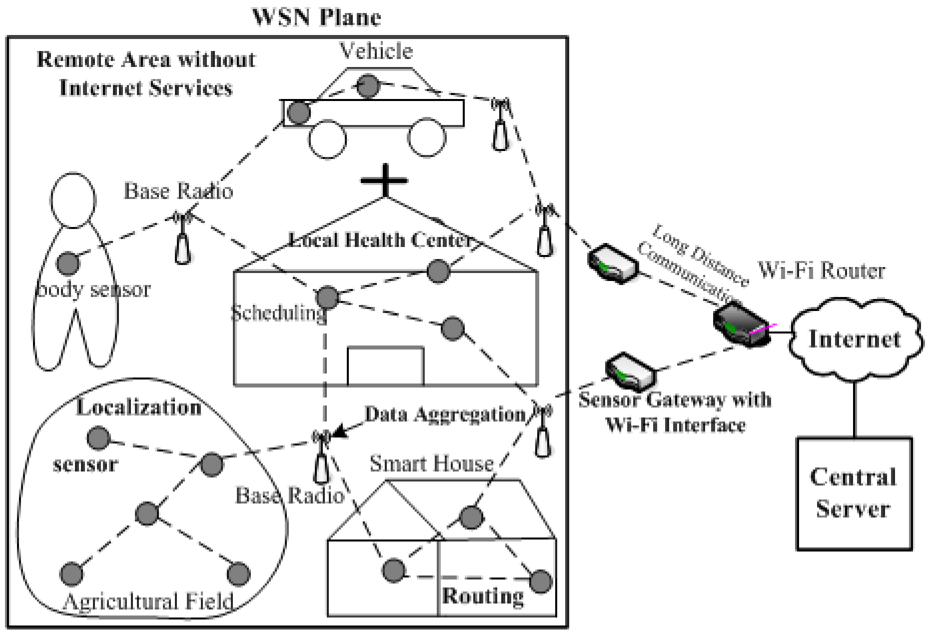

Figure 3.2: Architecture of Wireless Sensor Network plane

Figure 3.2 also demonstrates how these network management components work together to transmit data from the sensor nodes of this large scale WSN architecture to remote server through base station (BS) and Wi-Fi routers. Though we have shown each of these network management components at one place in the architecture (Figure 3.2) they will be implemented in most of the sensor nodes.

In Figure 3.3, we present the working flow of different components of the WSN management framework. Each of these component (e.g., sensor's localization, routing) are designed and implemented individually with its performance analysis and comparison with existing approaches. Finally, all these components are integrated in a single framework since they can be used as a complete solution in a WSN application. This is because, to best of my knowledge, no integrated framework consisting of localization, data scheduling, routing, and data aggregation approaches exists in the literature.

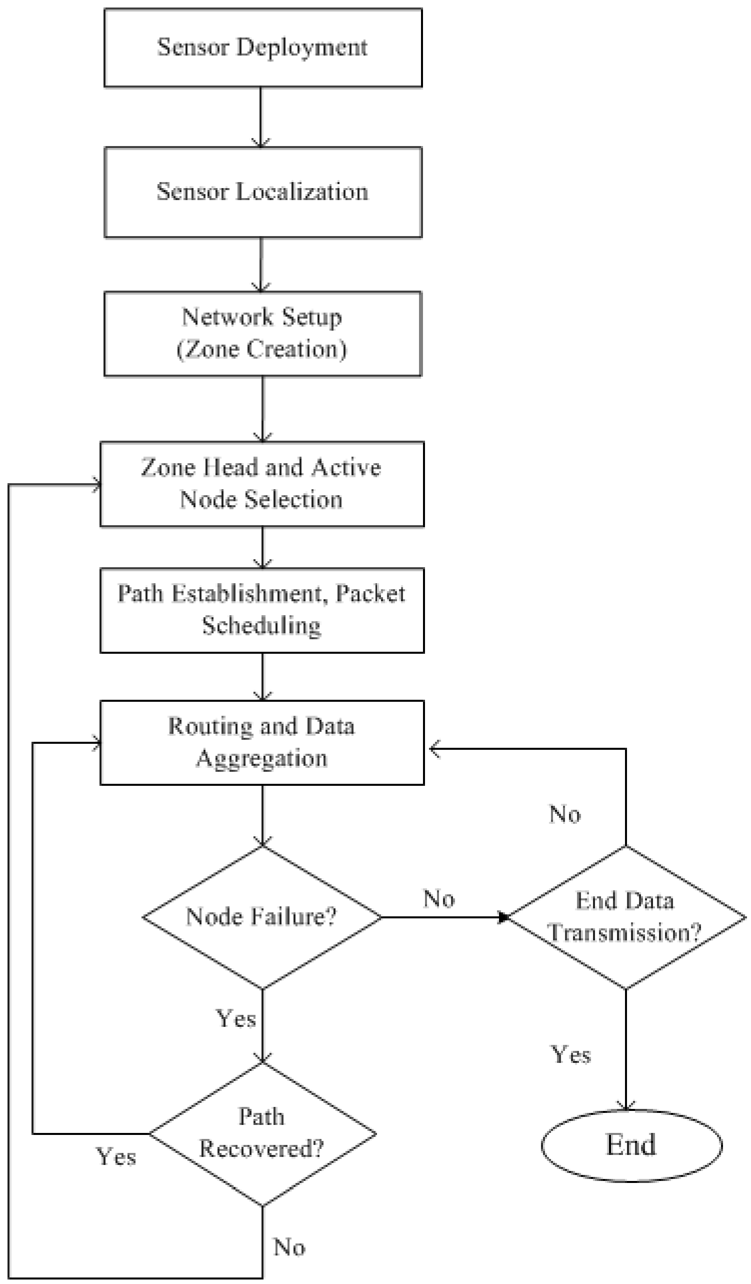

Figure 3.3: Flowchart of network management components of WSN

The first component in the flowchart of Figure 3.3 is sensors deployment. Based on the application types and deployment location, sensors can be deployed in two ways, which are as follows.

Manual Deployment

Sensors can be deployed manually if the application area is small, and human accessible. During manual deployment, coordinates of the sensors can either be assigned with respect to the network area using local map or be estimated using a localization approach.

Random Deployment

Sensors are deployed randomly if the application area is very large, remote, and human inaccessible such as mountain, forest, lake, and ocean. In random deployment, the locations of sensors are estimated using a localization approach. The proposed localization technique is presented in Chapter 4.

Once sensor's are localized, zones are created, zone head and active nodes are selected and data transmission paths among active nodes are created in the network setup phase of the proposed routing protocol (is presented in Chapter 5). Then a packet scheduling approach is introduced to determine the delivery order of different types of data packets, and achieve lower end-to-end data transmission delay (Chapter 6). Then routing of data (Chapter 5) and data aggregation (Chapter 7) are performed. The proposed routing protocol achieves reliability by introducing fault tolerance technique. The fault tolerance technique is presented in Chapter 5.

3.3 Assumptions and Terminologies

This section presents general assumptions and defines some important terminologies, and metrics that are used in designing the Wireless Sensor Network (WSN) management and control framework.

Assumptions

1. The WSN plane in the large scale multimodal architecture consists of a number of WSNs, for multiple applications. Thus, WSNs are considered as heterogeneous since different types of sensor nodes are used in different WSN based on the application requirements.

2. Each WSN is homogeneous, i.e., all the sensor nodes in the network have the same residual energy. In addition, those nodes have the same sensing range (R_s), and communication range (R_c).

3. The network is hybrid, i.e., it consists of both static and mobile nodes. Sensor nodes are mostly static. However, only a few mobile anchors nodes (defined in Section 3.3) are used to localize static un-localized nodes. The anchor nodes are attached to a small vehicle and thus, are mobile.

4. The network is modeled as a Unit Disk Graph (UDG), where any pair of sensor nodes communicates if the distance between this pair is less than the communication range. UDG is defined in Section 3.3.

5. The size of a data packet and special packet are fixed.

6. Multi-channel Medium Access Control (MAC) protocol is used.

7. Nodes in the network or zones form a virtual hierarchy, where they are considered to be located at different levels based on the number of hops each node is away from the base station (BS).

8. Sensor nodes use Time Division Multiple Access (TDMA) scheme to transmit data.

9. The shape of WSN area is square. BS divides the network into a number of square or rectangular zones. Then, we randomly deploy a number of sensor nodes in each zone. Thus, we have the control to deploy different number of nodes in different zones.

10. The radio pattern of the sensors is regular.

Unit Disk Graph

A sensor network is represented by a graph $G(V, E)$ where the term V represents a set of nodes (or vertices) and the term E represents a set of communication links (or edges). Two nodes, u and v, communicate with each other if they are within their communication range (R_c). This relation defines a common graph referred to as the Unit Disk Graph (UDG). However, since our proposed framework is in the context of sensor networks, we refer to the "double range property" [113] defined as $R_c = n \times R_s$, where R_s refers to the sensing range, and $n \geq 2$. In simulation, we represent the network using UDG.

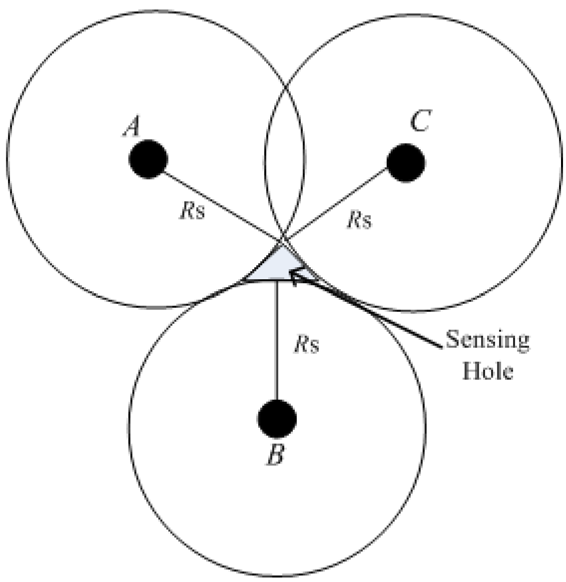

Figure 3.4: Sensing hole

Sensing Hole

Sensing hole is an area or point of the network, which is not covered by any sensor node. In other words, sensing hole is the area that is not sensed by any sensor node. Figure 3.4 illustrates a sensing hole, the small colored triangular area that is not covered by the nodes A, B, and C, where R_s is the sensing range of node A, B, and C.

Anchor Node

The nodes which know their locations or co-ordinates are called anchor nodes. They are integrated with Global Positioning System (GPS) or deployed manually to get their locations. Un-localized nodes in the network can estimate their locations using anchor nodes.

Sensors localization is a very import component of the proposed framework. Though we measure the efficiency of the complete WSN management framework we also measure the performance of the localization approach used in the WSN framework, and compare with existing approaches in terms of localization accuracy (or error), and localization energy consumptions. These two metrics are defined as follows.

Localization Accuracy

Localization accuracy is defined as the absolute difference between the actual positions of a node, i.e., the position obtained via Global Position System (GPS) to its estimated position that is obtained via the proposed or existing localization methods.

Localization Energy Consumptions

Sensor's localization energy consumptions is defined as the total localization energy consumptions (in Joules), which include the energy consumptions of anchor nodes to broadcast their positional information, and energy consumptions of both anchor and un-localized nodes for transmitting a large number of messages to identify the number of neighboring nodes.

Similarly, we evaluate and measure the performance of routing protocol, and data aggregation approaches in terms of network energy consumptions, and network lifetime.

Energy Consumptions

Energy consumptions of a node x in routing, and data aggregation is defined as the total energy consumptions (in Joule) of x for sending, and receiving data and special packets and aggregating data packets. Nodes that reside in the furthest zones from the BS or at the lowest level only send data and special packets. Intermediate nodes that reside between the furthest zones and BS receive data and special packets, aggregate data packets and then send the aggregated data to the active nodes at upper levels. However, we measure the performance of a network management approach in terms of the total network energy consumptions, which is the sum of energy consumptions of all nodes.

Network Lifetime

Network lifetime is defined as the total remaining energy of all sensor nodes of the network over a certain period of time. Let us assume that the initial network energy or energy of all nodes is equal to E_i. We estimate the energy consumptions of all sensor nodes in the network for routing and aggregating data for a certain period of time and denote it as E_t. Then, the network lifetime is estimated as $E_i - E_t$, the difference between E_t, and E_i.

Fault Tolerance

Fault tolerance ensures the continuing operation of WSN even in case of nodes failure. It reduces packet loss ratio by creating alternative paths for any established path between a source node and BS. Though most existing network management components of WSN are not fault tolerant our proposed framework will be designed with fault tolerant routing, and data aggregation approaches to detect the failure of nodes, and establish alternative paths by discarding the failed node from the faulty path. We measure fault tolerance in terms of $numPacketLoss$, number of packets loss over a certain period of time, which is actually the difference between the number of packets ($numPacketWithFT$) transmitted using fault tolerance mechanism, and number of packets ($numPacketWithoutFT$) transmitted without any fault tolerance mechanism. Equation 3.1 denotes the number of packet loss.

$$numPacketLoss = numPacketWithFT - numPacketWithoutFT \tag{3.1}$$

Data Propagation Time or End-to-End Delay

Data propagation time or end-to-end delay is defined as the total time that is required to transmit sensors' data from the source node to end users. However, in our proposed routing and data aggregation approach, we define the time that is required to transmit data from leaf nodes or nodes at the farthest zones to BS as end-to-end data transmission delay.

3.4 Summary

In this chapter, we presented a large scale wireless network architecture that entails three overlay networks: Wireless Sensor Network (WSN), Wi-Fi, and WiMAX. Then, we presented the architecture of the WSN management framework that comprises sensors localization, routing protocol, data scheduling, and data aggregation approaches. We also presented general assumptions, and defined terminologies and performance metrics. In the next chapter, we will present our proposed range-free, and energy efficient localization approach, namely RELMA with its performance analysis, and evaluation.

Chapter 4: Sensor Localization

Most localization techniques in the literature are designed for small scale Wireless Sensor Networks (WSNs) and any particular application. Some of these localization techniques use powerful sensor nodes equipped with Global Positioning Systems (GPS) to estimate their positions. However, GPS is expensive to equip with all nodes, and cannot function for indoor applications since there is no direct Line of Sight (LOS) to the satellite. To reduce the cost, few nodes are equipped with GPS that are known as anchor nodes. These nodes broadcast their positions through beacons. The un-localized nodes estimate their positions based on these positional information of anchor nodes. However, these algorithms are mostly range-based and use special hardware or techniques such as Received Signal Strength Indicator (RSSI), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA) [124] to measure distance using static anchor nodes that are deployed at some predefined locations. On the other hand, range free algorithms exploit sensor's connectivity information instead of using energy expensive distance estimation. Hence, range free algorithms are energy efficient as compared to range-based algorithms. However, range free algorithms mostly use a large number of static anchor nodes. In these algorithms, a node is also considered as an anchor node whenever, its position is estimated. If there is any localization error for estimating the position of a node, the error is propagated to localize remaining nodes that results a large average localization error. Thus, we introduce a new localization method referred to as the Range-free Energy efficient, Localization technique using Mobile Anchor (RELMA) [49].

The remainder of this chapter is organized as follows. Section 4.1 presents the working principle of RELMA. Section 4.2 presents the theoretical analysis of energy consumptions of RELMA approaches. Section 4.3 presents the simulation setup, and results. Analysis and explanations of the performance of RELMA methods are discussed in Section 4.3.3. Finally, Section 4.4 concludes this chapter with future research ideas.

4.1 Working Principle of RELMA

We assume that all sensors nodes in a WSN have the same sensing, R_s, and communication ranges, R_c. Two nodes can communicate with each other if their sensing circles intersect by at least one point. For instance, in Figure 4.1, $R_s(A)$ and $R_s(B)$ are the sensing radii, and $R_c(A)$ and $R_c(B)$ are the communication radii of nodes A and B, respectively, which can communicate with each other since their sensing circles intersect by one point. However, as a more specific case in RELMA, each node communicates with nodes that are inside its sensing circle to reduce energy consumptions in localization.

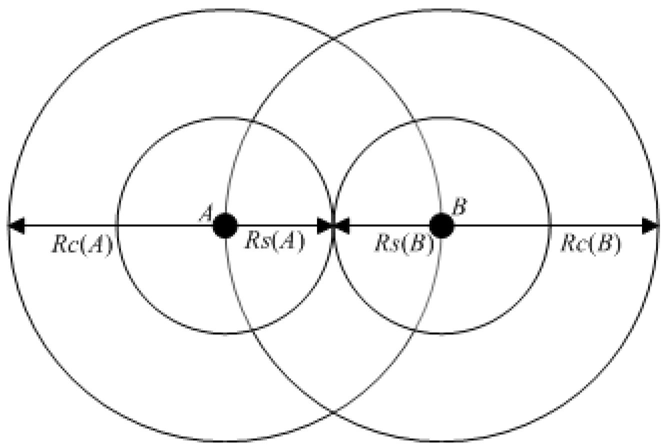

Figure 4.1: Sensing and communication circles of two nodes A and B

Once the nodes are deployed randomly in the network area they localize themselves using the location information of anchor nodes. We deploy a few mobile anchor nodes if the network is large. In a small network (e.g., network size in our simulation model), we deploy only one anchor node that is attached to a small vehicle and is mobile. The anchor node moves in a low velocity in the network and broadcasts a message containing its position. Whenever a mobile anchor node broadcasts the position from its current location RELMA assumes that a static node exists at that position for the ease of computation. If an un-localized node P receives three anchor positions it starts the localization process. Though P has three or more anchor positions from the mobile anchor RELMA does not include range-based distance estimation such as trilateration rather it exploits the interconnectivity of P with these virtual anchor nodes/positions using sensing range.

Once the node P receives three positional information from mobile anchor nodes P estimates its position either using RELMA Method 1 or RELMA Method 2, which are presented in the following sections.

4.1.1 RELMA Method 1

The RELMA Method 1 uses the following technique to estimate the position of an un-localized node.

Sensing circles of three anchor nodes or positions intersect at three points and form the intersected region R (triangle). Using this method, the coordinates of the three intersected points of R are calculated. Then the coordinate of $P(X_P, Y_P)$ is estimated using Equations 4.1 and 4.2 as the Centroid of R. The intersected region R of the sensing circles of three anchor positions is much smaller than the intersected region of their communication circles. Hence, the estimated localization error is expected to be less. Figure 4.2 illustrates the location estimation using the RELMA Method 1.

$$X_P = \frac{(X_1 + X_2 + X_3)}{3} \tag{4.1}$$

$$Y_P = \frac{(Y_1 + Y_2 + Y_3)}{3} \tag{4.2}$$

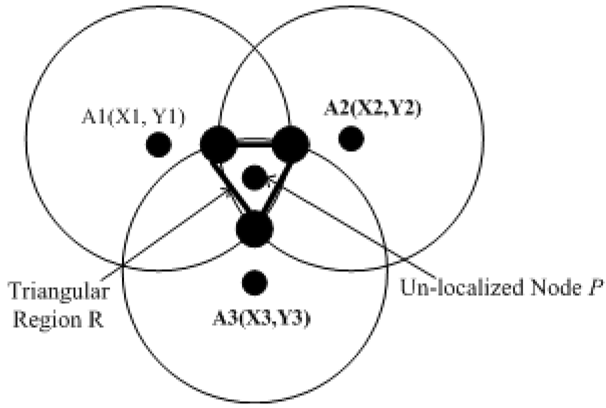

Figure 4.2: Centroid of a triangle formed by three intersected points (RELMA Method 1)

Since the unknown node P resides in the region R any position of P in R is expected to result in a localization error. The localization approach is more effective if the localization error is very small to be ignored. However, to estimate the localization error we assume that the actual position of node P is known (in our simulation). Thus, we can estimate the localization error in node P by subtracting its estimated position from the actual position.

4.1.2 RELMA Method 2

In RELMA Method 2, the unknown node P estimates its position using the neighboring set and distance approaches [66]. A set of neighboring nodes, which are in the sensing range of a node is known as its Neighboring Set (NS).

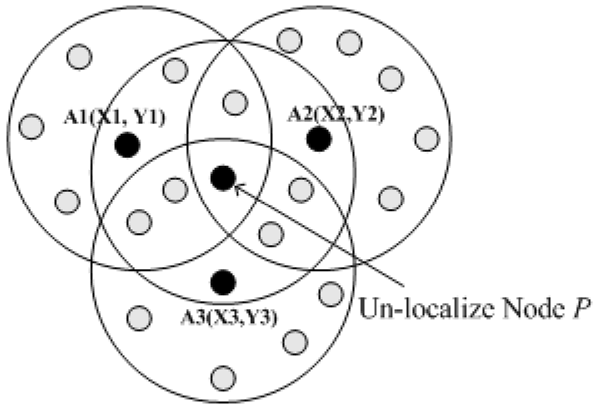

Figure 4.3: Neighboring set and distance localization approach (RELMA Method 2)

Neighboring Distance (ND_{PA_i}) of a node P with its neighbor A_i (i.e., mobile anchor node) is the ratio of the number of neighboring nodes for A_i (NS_{A_i}) to the number of nodes resulted from $NS_P \cap NS_{A_i}$, see the following equation.

$$ND_{PA_i} = \frac{|NS_{A_i}|}{|NS_P \cap NS_{A_i}|} \tag{4.3}$$

Where $A_i, 1 \leq \ i \leq 3$ are three positions of the mobile anchor node that moved through the neighborhood of P. For instance, $|NS_{A_1}| = 8$, $|NS_P| = 10$, $|NS_P \cap NS_{A_1}| = 5$ and so, $ND_{PA_i} = \frac{8}{5}$ in Figure 4.3. Now the coordinate of P is estimated using the following equations.

$$X_P = \frac{\frac{X_{A_1}}{ND_{PA_1}} + \frac{X_{A_2}}{ND_{PA_2}} + \frac{X_{A_3}}{ND_{PA_3}}}{\frac{1}{ND_{PA_1}} + \frac{1}{ND_{PA_2}} + \frac{1}{ND_{PA_3}}} \tag{4.4}$$

$$Y_P = \frac{\frac{Y_{A_1}}{ND_{PA_1}} + \frac{Y_{A_2}}{ND_{PA_2}} + \frac{Y_{A_3}}{ND_{PA_3}}}{\frac{1}{ND_{PA_1}} + \frac{1}{ND_{PA_2}} + \frac{1}{ND_{PA_3}}} \tag{4.5}$$

Where, (X_{A_1}, Y_{A_1}),(X_{A_2}, Y_{A_2}), (X_{A_3}, Y_{A_3}) are coordinates of three positions of the mobile anchor node, and (X_P, Y_P) is the coordinate of P.

For densely deployed sensor network, it is expected that the size of NS for each of the anchor nodes or positions and the un-located node P is high and for a small intersected region between an anchor node and P, the number of nodes that are contained in both the NS of the anchor node and P will be small. This results a high value of ND for each of the anchor nodes. On the contrary, for sparse sensor deployment, the value of ND will be small. When the value of ND for each of three anchor nodes for locating P is high, the value of the nominators of Equations 4.4 and 4.5 will be lower than the value that is expected for getting more accurate coordinate of the un-located node P. Hence, denominator that will be less for the higher value of ND will normalize the coordinate of P. Similarly, for the lower value of ND, the value of the nominator and denominator will be high that again normalizes the coordinates of P.

We can estimate the localization error in node P by subtracting its estimated position using RELMA Method 2 from its actual position. Again we assume that the actual position of P is known to estimate the localization error through simulation. Based on the localization error in both approaches we can consider the approach (either RELMA Method 1 or RELMA Method 2) as the most appropriate one, which has the lowest localization error.

4.2 Theoretical Analysis

We use either RELMA Method 1 or RELMA Method 2 to estimate the location of a node. Hence, we analyze the performance of RELMA Method 1, and RELMA Method 2 in terms of energy consumptions of the sensor nodes.

4.2.1 Energy Model

Let E_{Tx} and E_{Rx} represent energy consumptions of an un-localized node for transmitting and receiving data of size, *datasize*, to/from another node at distance d and are denoted as

$$E_{Tx} = datasize \times \epsilon_{data} + datasize \times d^2 \times \epsilon_{air} \tag{4.6}$$

$$E_{Rx} = datasize \times \epsilon_{data} \tag{4.7}$$

In Equations 4.6 and 4.7, ϵ_{data} and ϵ_{air} represent the energy spent in transmitter electronics circuitry, and in radio frequency (RF) amplifiers for propagation loss, respectively.

4.2.2 Localization Energy Consumptions

In this section, we calculate the energy consumptions of sensor nodes for RELMA Method 1 and Method 2.

RELMA Method 1

In RELMA Method 1, an un-localized node P receives messages with positional information of at least three mobile anchor nodes or positions A_i, where, $1 \leq i \leq 3$. P is at $d_i(A_iP) \leq R_S(P)$ distance apart from A_i where, $R_S(P)$ is the sensing rage of P. Thus, the total transmitting energy consumptions, $E_{Tx1}(A)$ for the anchor nodes A_i can be estimated as

$$E_{Tx1}(A) \geq \sum_{i=1}^{3}(datasize1_i \times \epsilon_{data} + datasize1_i \times d_i^2(A_iP) \times \epsilon_{air}) \tag{4.8}$$

Similarly, the receiving energy consumptions, $E_{Rx1}(P)$ of node P will be

$$E_{Rx1}(P) \geq \sum_{i=1}^{3}(datasize1_i \times \epsilon_{data}) \tag{4.9}$$

Where, $datasize1_i$ is the size of a message that is transmitted from the i-th anchor node to the un-localized node P.

We assume that the size of data packet that contains the positional information of all three anchors is equal to $datasize1$ (i.e., fixed size). Thus, the total transmitting energy consumptions, E_{Tx1}, and receiving energy consumptions, E_{Rx1} to localize n number of un-located nodes in the network using RELMA Method 1 can be expressed as

$$E_{Tx1} \geq \sum_{j=1}^{n} \sum_{i=1}^{3} (datasize1 \times \epsilon_{data} + datasize1 \times d_{ji}^{2}(A_{ji}P_{j}) \times \epsilon_{air} \tag{4.10}$$

$$E_{Rx1} \geq 3 \times n \times datasize1 \times \epsilon_{data} \tag{4.11}$$

RELMA Method 2

In RELMA Method 2, anchor nodes A_j where, $1 \leq j \leq 3$ and un-located node P broadcast messages to identify the number of neighboring nodes. The nodes Y_i, which are within the sensing range of A_j and/or P replies. Let us assume that each of three anchor nodes/positions, A_j receives message from $m_{A_j} = |NS_{A_j}|$ neighboring nodes. Each of the neighboring nodes Y_i of A_j is at distance $d_{ji}(Y_iA_j) \leq R_s(A_j)$ apart from the anchor node A_j, where $1 \leq i \leq m_{A_j}$. Thus, the total transmitting energy consumptions, $E_{Tx2}(A_j)$, and $E_{Tx2}(Y_{A_j})$ of the anchor nodes A_j, and all neighboring nodes Y_i of A_j will be

$$E_{Tx2}(A_j) \geq \sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} (datasize2_{A_{ji}} \times \epsilon_{data} + datasize2_{A_{ji}} \times d_{ji}^{2}(A_Y) \times \epsilon_{air} \tag{4.12}$$

$$E_{Tx2}(Y_{A_j}) \geq \sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} (datasize2_{Y_{ji}} \times \epsilon_{data} + datasize2_{Y_{ji}} \times d_{ji}^{2}(Y_A) \times \epsilon_{air} \tag{4.13}$$

In Equations 4.12 and 4.13, $datasize2_{A_{ji}}$ is the size of the message that is transmitted from j-th anchor position to i-th neighboring node and $datasize2_{Y_{ji}}$ is the size of reply message that is transmitted from i-th neighboring node to j-th anchor node.

Let us assume that $datasize2_{A_{ji}} = datasize2_{Y_{ji}} = datasize2$ and $E_{Tx2}(A_j) \approx E_{Tx2}(Y_j)$.

Therefore, the total transmitting energy consumptions to find neighbors of three anchor nodes/positions (that is, required to localize node P) can be expressed as

$$E_{Tx2}(AY) \geq 2 \times \sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} (datasize2 \times \epsilon_{data} + datasize2 \times d_{ji}^2(AY) \times \epsilon_{air}) \quad (4.14)$$

Similarly, Equation 4.15 estimates of the total receiving energy consumptions to find the neighbors of three anchor nodes or positions.

$$E_{Rx2}(AY) \geq 2 \times \sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} (datasize2 \times \epsilon_{data}) \quad (4.15)$$

Similarly, the total energy consumptions, $E_{T_{x2}}(P)$, and $E_{T_{x2}}(Y_P)$ of un-located node P and all neighboring nodes Y_i of P will be

$$E_{Tx2}(P) \geq \sum_{i=1}^{m_P} (datasize2_{P_i} \times \epsilon_{data} + datasize2_{P_i} \times d_i^2(Y_P) \times \epsilon_{air}) \quad (4.16)$$

$$E_{Tx2}(Y_P) \geq \sum_{i=1}^{m_P} (datasize2_{Y_i} \times \epsilon_{data} + datasize2_{Y_i} \times d_i^2(Y_P) \times \epsilon_{air}) \quad (4.17)$$

In Equations 4.16 and 4.17, $datasize2_{P_i}$ is the size of the message that is transmitted from P to i-th neighbor and $datasize2_{Y_i}$ is the size of the reply message that is transmitted from i-th neighbor node to P

Again, we assume that $datasize2_{P_i} = datasize2_{Y_i} = datasize2$ and $E_{Tx2}(P) \approx E_{Tx2}(Y_P)$

Therefore, the total transmitting and receiving energy consumptions for the unknown node P and its neighbors can be estimated as

$$E_{Tx2}(PY) \geq 2 \times \sum_{i=1}^{m_P} datasize2 \times \epsilon_{data} + datasize2 \times d_i^2(Y_P) \times \epsilon_{air} \tag{4.18}$$

$$E_{Rx2}(PY) \geq 2 \times \sum_{i=1}^{m_P} datasize2 \times \epsilon_{data} \tag{4.19}$$

Without loss of generality, the total transmitting energy consumptions E_{Tx2} and receiving energy consumptions E_{Rx2} that are required to localize sensor nodes in the network using RELMA Method 2 can be expressed as follow.

$$\begin{aligned} E_{Tx2} \geq 2 \times n \times (&\sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} datasize2 \times \epsilon_{data} + datasize2 \times d_{ji}^2(AY) \times \epsilon_{air} \\ &+ \sum_{i=1}^{m_P} datasize2 \times \epsilon_{data} + datasize2 \times d_i^2(Y_P) \times \epsilon_{air}) \end{aligned} \tag{4.20}$$

$$E_{Rx2} \geq 2 \times n \times (\sum_{j=1}^{3} \sum_{i=1}^{m_{A_j}} datasize2 \times \epsilon_{data} + \sum_{i=1}^{m_P} datasize2 \times \epsilon_{data}) \tag{4.21}$$

Comparison of RELMA Method 1 and Method 2

Now, comparing Equation 4.10 with Equation 4.20 and Equation 4.11 with Equation 4.21 we can infer that

$$E_{Tx1} < E_{Tx2} \tag{4.22}$$

$$E_{Rx1} < E_{Rx2} \tag{4.23}$$

4.2.2.1 Neighbor-information-based Localization System (NBLS)

Neighbor-information-Based Localization System (NBLS) is divided into three phases. In the first phase, all nodes receive their neighboring nodes information, and use an improved DV-hop method to estimate their locations. The nodes that can estimate their positions are known as unexamined nodes. In the second phase, the unexamined nodes that have localization error less than a threshold value are selected as quasi-anchor nodes. Unexamined nodes with error greater than the threshold value are selected as further estimated nodes. In the third phase, locations are estimated for further estimated nodes [66].

Since RELMA Method 2 is based on the neighboring nodes information (but uses sensing range instead of communication range that NBLS uses), RELMA Method 2 is a part of the third phase of NBLS. Thus, we can infer that the transmitting, and receiving energy consumptions for localizing n sensor nodes in NBLS will be

$$E_{Tx\_NBLS} > E_{Tx2} \tag{4.24}$$

$$E_{Rx\_NBLS} > E_{Rx2} \tag{4.25}$$

Finally, observing Equations 4.22 - 4.25 we can conclude that

$$E_{Tx1} < E_{Tx2} < E_{Tx\_NBLS} \tag{4.26}$$

$$E_{Rx1} < E_{Rx2} < E_{Rx\_NBLS} \tag{4.27}$$

4.3 Performance Evaluation

In this section, we present performance metrics, and simulation results.

4.3.1 Performance Metrics

We evaluate the performance of the proposed RELMA localization approach in terms of localization accuracy (or error), and localization energy consumptions. Both of these metrics are defined in Section 3.3 of Chapter 3.

4.3.2 Simulation Setup and Results

We simulate the proposed RELMA localization approach using C programming language. The reason we choose C programming language to build our simulator are as follows.

- Most existing network simulators, such as ns-2, and opnet are built using C/C++.
- We also simulate other proposed network management approaches, such as routing protocols, and data aggregation approaches using C programming language (are presented in Chapters 5 and 7). We find that existing approaches for sensors localization, routing, data scheduling, and data aggregation that we compare with the proposed approach are simulated using various simulators. Thus, we use C programming language to build the simulator rather than to use different simulators for each of the network management approaches.
- Even if we used different simulators for different network management approaches we had to use anyone of these simulators to integrate all these approaches as a single framework. Hence, it would be better to use C programming language as it is basis to build simulators for each of the approaches and also for the integrated framework.

We use randomly connected Unit Disk Graphs (UDGs) on an area of 100 meters $\times$ 100 meters as a network simulation model. We list the simulation parameters, and their respective values in Table 4.1.

Table 4.1: Simulation parameters and their respective values

Parameter	**Value**
Network Size	$100m \times 100m$
Number of Nodes	Maximum 180
Number of Zones	4
Data packet size	256 bits
Transmission Energy Consumptions	$50nJoule/bit$
Energy Consumption in free space or air	$0.01nJoule/bit/m^2$
Initial Node Energy	2 Joule
Mobile Anchor Velocity	$1 \sim 5$ meters/minute
Sensing Range (R_s)	$10 \sim 20$ meters
R_c/R_s constant	$2 \sim 4$

Firstly, we check whether data or samples collected from sensors through simulation are normally distributed. If so, we perform student's t-test or ANOVA test at 90 - 95% confidence level to check whether the systems are identical or not. On an average, we take 25 - 100 samples at each data point in localization and other network management approaches and take the mean of these samples to perform the t-test at 90 - 95% confidence level. We follow the same process for the simulation results of other approaches such as routing and data aggregation.

In the following subsections, we present the simulation results.

4.3.2.1 Varying Number of Nodes

We vary the number of nodes between 100 and 180. We set the transmission range (R_c), and the sensing range (R_s) to 30 meters, and 15 meters, respectively. For the small network area in our simulation, we use a single mobile anchor node, which is attached to a toy car and moves at the low velocity, 3 meters/minute. Thus, an un-localized node gets enough time to receive the positional information of anchor nodes, and computes its position.

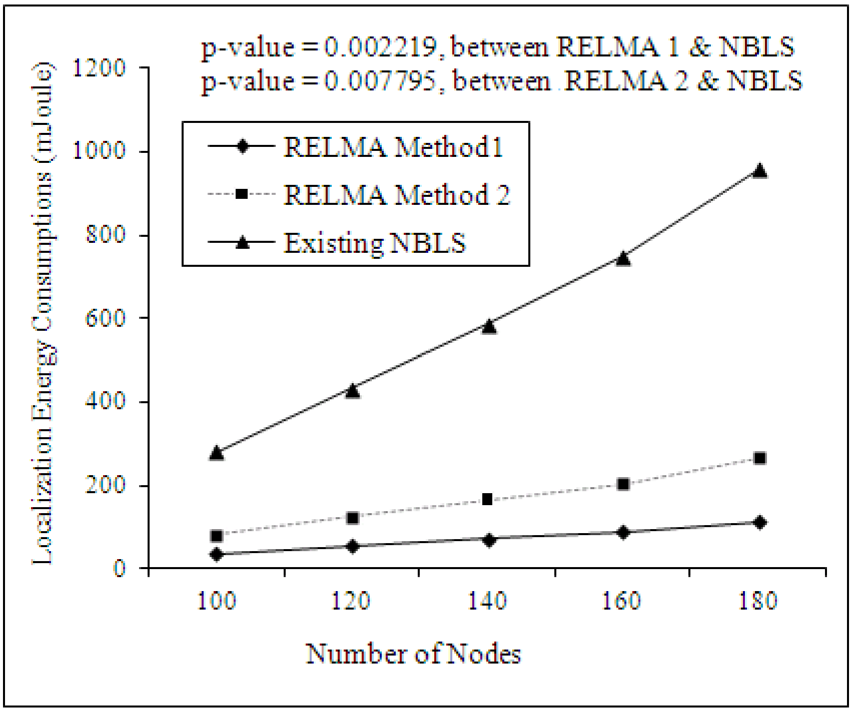

Figure 4.4: Comparison of localization energy consumptions of RELMA Method 1 and Method 2 with existing NBLS varying number of nodes

Figure 4.4 illustrates that both of RELMA Method 1 and RELMA Method 2 have better performance than the existing NBLS approach in terms of localization energy consumptions. This is because the number of anchor nodes in RELMA approaches is lower than that in NBLS. These anchor nodes transmit beacons with their positional information to un-localized nodes in RELMA whenever the anchor nodes are within the sensing range, R_s of the un-localized nodes. In NBLS, anchors nodes transmit beacons whenever they are within the communication range, R_c of un-localized nodes. We know from the "double range property" [113] that the sensing range is much shorter than the communication range since $R_c = n \times R_s$, where $n \geq 2$. However, RELMA Method 1 is much more energy efficient than RELMA Method 2. This is because anchor nodes and un-localized nodes transmit a large number of messages to identify their neighboring set in RELMA Method 2, which is not required in RELMA Method 1. We also perform student's t-test to validate the simulation results at 95% confidence level. Figure 4.4 shows that the p-values are equal to 0.002219 between RELMA Method 1 and NBLS, and 0.007795 between RELMA Method 2 and NBLS, which validate our claim.

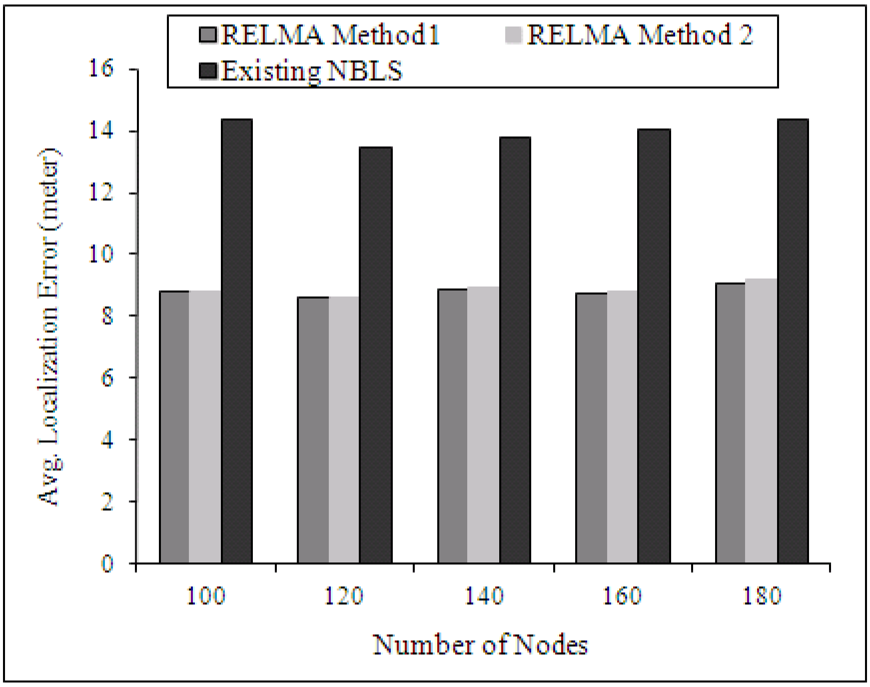

Figure 4.5: Comparison of localization error of RELMA Method 1 and Method 2 with existing NBLS varying number of nodes

Similarly, Figure 4.5 demonstrates that both of RELMA Method 1 and RELMA Method 2 outperform the existing NBLS approach in terms of the average localization error for varying the number of nodes. This is because whenever a node, X is localized in NBLS it is considered as an anchor node. Thus, localization error in X will be propagated to the rest of un-localized nodes that results in a large localization error. This is not the case for the proposed RELMA approaches. Interestingly, both of the proposed RELMA Method 1, and RELMA Method 2 have no significant difference in term of localization accuracy. Moreover, Figure 4.5 shows that the average localization error remains almost the same for each approach even if we vary the number of nodes since $avgLocError = \frac{totalError}{totalNumOfNodes}$.

4.3.2.2 Varying Sensing Range, R_s

In this section, we present simulation results when we vary sensing range, R_s between 10 and 20 meters. We set the number of sensors, zones, velocity of mobile anchor, and R_c/R_s constant to 100, 4, 4 meters/second and 2, respectively.

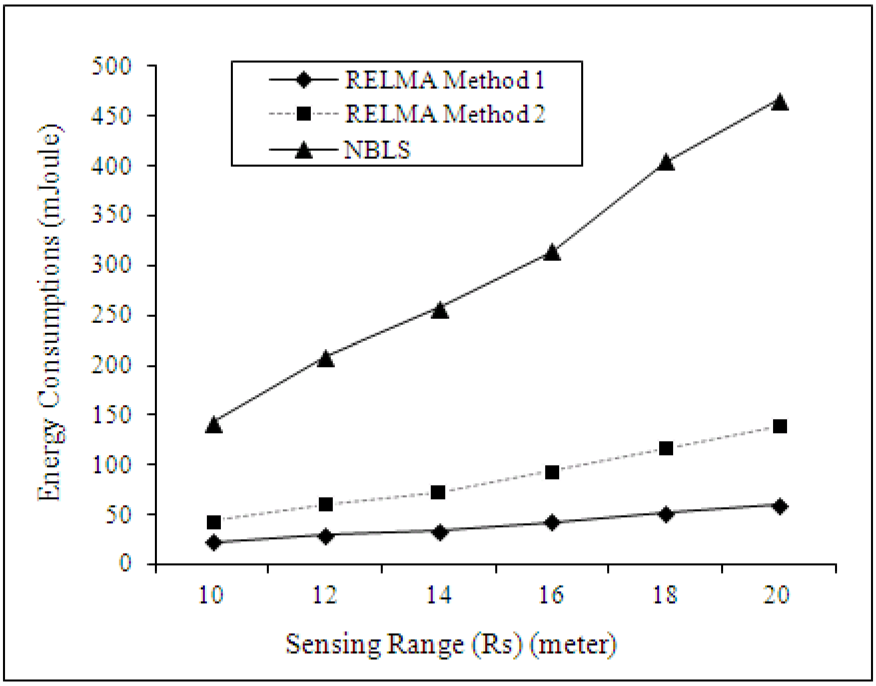

Figure 4.6: Comparison of localization energy consumptions of RELMA Method 1 and Method 2 with existing NBLS varying sensing range R_s

Figure 4.6 illustrates that the localization energy consumptions of both RELMA

Method 1, and RELMA Method 2 are much lower than that of NBLS. The localization energy consumptions in RELMA Method 2 is much more than that in RELMA Method 1 since in RELMA method 2, each node transmits a large number of messages to compute its Neighboring Set (NS) and Neighboring Distance (ND). We also find in Figure 4.7 that localization error in NBLS is much larger than that in both RELMA Method 1, and RELMA Method 2 because in NBLS, an un-localized node, P receives a message with the positional information of an anchor node whenever the anchor node is within the communication range, R_c of P. The proposed RELMA localization approach uses sensing range, R_s for this purpose. These results are also validated through statistical analysis (student't t-test).

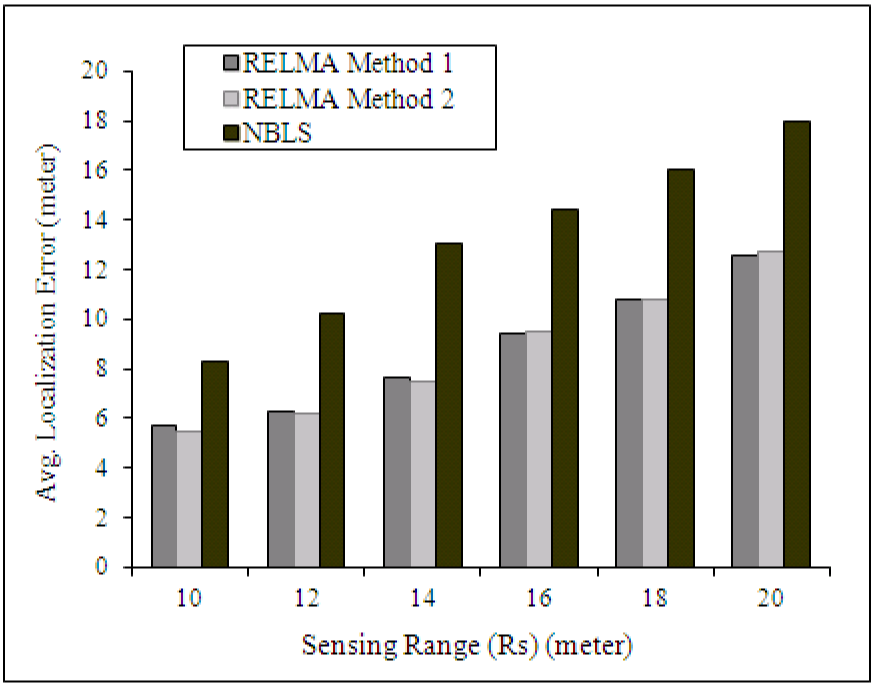

Figure 4.7: Comparison of localization error of RELMA Method 1 and Method 2 with existing NBLS varying sensing range R_s

However, the localization error increases in both RELMA approaches for varying R_s. This is because the intersected region where the un-localized node belongs to becomes larger in RELMA. Communication range, R_c also increases for using the "double range property" [113] in NBLS.

4.3.2.3 Comparison of RELMA Method 1 and Method 2 Varying Velocity of Anchor Nodes

In this section, we present simulation results of RELMA Method 1 and RELMA Method 2 for varying the velocity of mobile anchor node between 1 and 5 meters/second. We set the number of sensors, zones, sensing range R_s, and constant n to 100 nodes, 4 zones, 15 meters and 2, respectively. Figure 4.8 illustrates that RELMA Method 1 consumes less energy than that in RELMA Method 2. Figure 4.9 also demonstrates that both RELMA Method 1 and Method 2 have the similar localization accuracy for varying the velocity of mobile anchor node. We have already explained the possible reasons for these performance differences in Section 4.3.2.1.

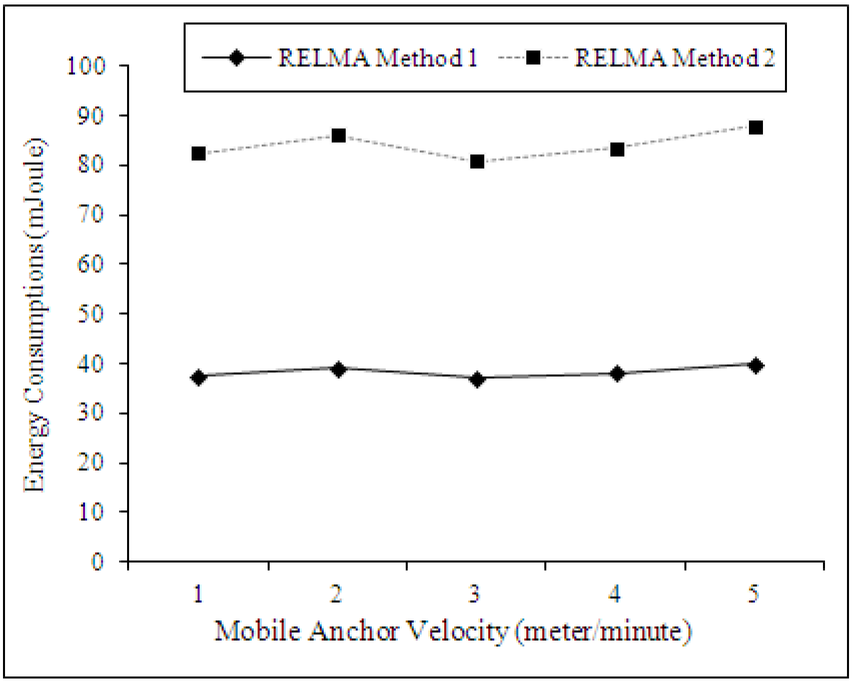

Figure 4.8: Comparison of energy consumptions of RELMA Method 1 and Method 2 with NBLS varying the moving speed of anchor node

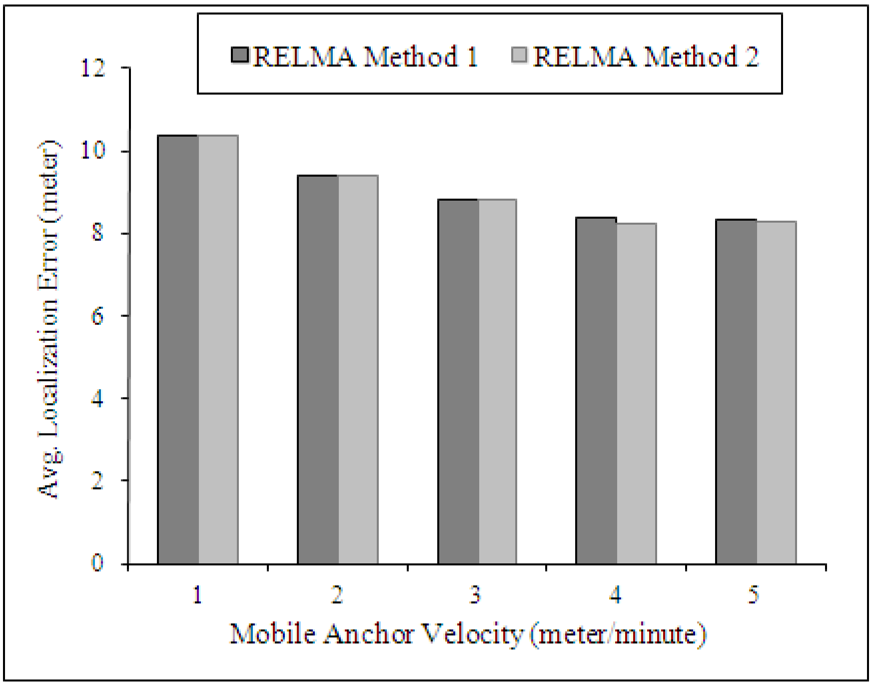

Figure 4.9: Comparison of localization error of RELMA Method 1 and Method 2 with NBLS varying the moving speed of anchor node

However, energy consumptions, and localization error remain almost the same in each approach even if the velocity of the mobile anchor node increases. This is because the movement of the anchor node can be considered as slow, for the velocity in the range between 1 and 5 meters/minute.

4.3.3 Discussion

The idea behinds RELMA Method 2 is that we will have a proper estimate of the location of unknown node P when the number of nodes in the neighboring set(NS) of each of the three anchor positions is almost equal. However, it is more unlikely to have nodes equally distributed to the neighborhood of each sensor node in a small scale sensor network. Each node might have a large number of neighboring nodes in large scale sensor networks that results in almost equal Neighboring Distance (ND) and better localization accuracy. In RELMA Method 1, the area, R in Figure 4.2 where an unknown node resides is very small since the sensing range, R_s of a node is much shorter than the communication range, R_c. Hence, taking the Centroid position of that region is also expected to provide an accurate location estimation in RELMA Method 1. However, RELMA Method 2 requires a large number of messages transmissions for calculating the NS and ND of a node. Hence, RELMA Method 2 is less energy efficient as compared to RELMA Method 1. On the other hand, on top of calculating the NS, and ND, a large number of communications takes place for selecting anchor and quasi-anchor nodes in NBLS [66] that consumes much energy of a sensor node. Moreover, the NBLS method uses R_c in a global map that results in a more localization error.

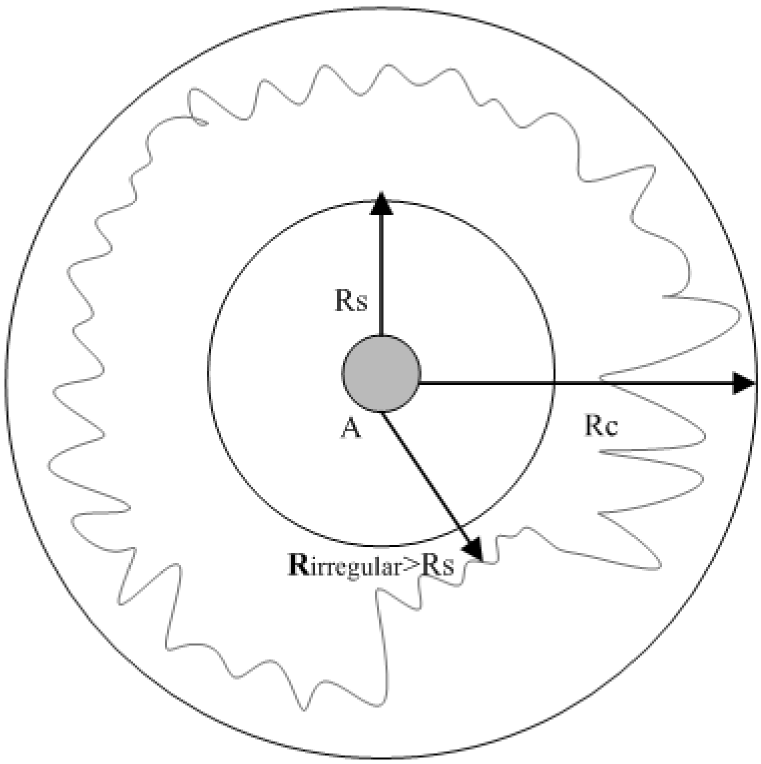

Figure 4.10: Irregular radio pattern

Sensing range, R_s is used in both RELMA Method 1, and RELMA Method 2 instead of the communication range, R_c that reduces the energy consumptions of sensor nodes in RELMA. We make an assumption that the radio pattern of our simulated sensor network is regular. However, it is more unlikely that R_c of a node becomes shorter than R_s due to the irregular radio. Hence, we can assume that RELMA can maintain the communication even if the irregular radio pattern [129] arises. Figure 4.10 clarifies the claim regarding the proposed RELMA localization approach since the sensing range, $R_s < R_{irregular}$, communication range of irregular radio in all directions.

4.4 Summary

In this chapter, we introduced a Range free Energy efficient Localization approach using Mobile Anchor (RELMA) for a large scale Wireless Sensor Network (WSN). RELMA uses sensing range for each pair of nodes to communication while most existing approaches consider communication range. In addition, RELMA uses only a few mobile anchor nodes in a local map. Theoretical analysis and simulation results demonstrated that RELMA is more energy efficient, and accurate as compared to NBLS [66] - an existing Neighboring information based localization approach. We evaluated the performance of RELMA Method 1, and RELMA Method 2 with NBLS and found that both methods have better performance than NBLS in terms of localization energy consumptions, and average localization error. However, both of RELMA Method 1 and Method 2 have similar performance in terms of localization accuracy but RELMA Method 1 is more energy efficient than RELMA Method 2. In future, we plan to compare the performance of RELMA approaches against other range free localization techniques and consider radio irregularity to measure its impact and performance. In the next chapter, we will present our proposed routing protocol for a large scale WSN with its performance analysis, and evaluation.

Chapter 5: Topology Control and Routing Protocol

Many of the existing routing protocols [2, 18, 42, 46] use energy inefficient tree structure, and dynamic flooding. These protocols use the number of hops as the length of data transmission path. However, the actual length of the shortest hop path can be longer than that of a larger hop path. Moreover, there are routing protocols [34] that use the global topological information for message forwarding where sensor nodes have limited resources such as memory size. Combining all these drawbacks leads to the following research question: how to construct a routing protocol that has the following interesting behaviors: 1) being efficient in terms of the power consumptions, and 2) using the local neighborhood's information for packet forwarding. Related to this problem is the topology control challenge for Wireless Sensor Network (WSN) where several algorithms are used [21, 29, 103]. However, the constructed topologies remove too many edges (i.e., wireless links). As a consequence, many shortest paths are discarded. Thus, the packet delivery speed is dramatically reduced. In addition, the power consumption for the source-destination paths is significantly increased. This introduces another challenge: how to maintain as much of the shortest paths as possible when new topologies are constructed. To the best our knowledge, there is no existing work that addresses all these challenges in one framework. Thus, we introduce a routing protocol referred to as the Cluster-based Routing and Topology

Control Approach (CRTCA) [47, 52, 53] for large scale Wireless Sensor Networks (WSNs). We incrementally integrate clustering, routing, and topology control approaches to construct this protocol.

This chapter is organized as follows. Section 5.1 presents the setup phase, i.e., zone creation, base station (BS) placement, node distribution, active nodes selections, and paths establishment among active nodes to the BS. Section 5.3 presents a new set of graphs referred to as the Mini Gabriel (MG) graphs [29, 81]. Sections 5.4, and 5.5 briefly present the steady phase (i.e., routing data), and fault tolerance technique, respectively. The simulation results based on an existing Gabriel Graph (GG) and a new set of graph, MG are presented in Section 5.6. We also perform experimentation (both mathematical analysis, and simulation) to evaluate and compare the performance of CRTCA with the existing standard clustering protocols, LEACH [38], and DSC [8]. Experimental results are presented in Section 5.7. Finally, Section 5.8 presents a summary of this chapter.

5.1 Zone Creation and Active Node Selection

In this section we present the necessary steps to setup the topology and network.

5.1.1 Zone Creation and Node Distribution

The network area where sensors are deployed can be of any shape, e.g., polygon, circle, triangle. However, these shapes can be circumscribed into a square as is illustrated in Figure 5.1. Thus, we assume that the shape of the network is square.

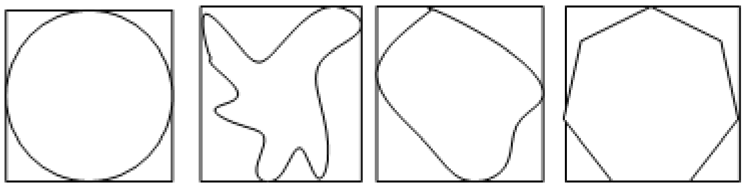

Figure 5.1: Different shapes of a network area

Once sensors are localized, the network area is divided into a number of square zones. The base station (BS) assigns the sensor nodes to different zones based on their geometric positions.

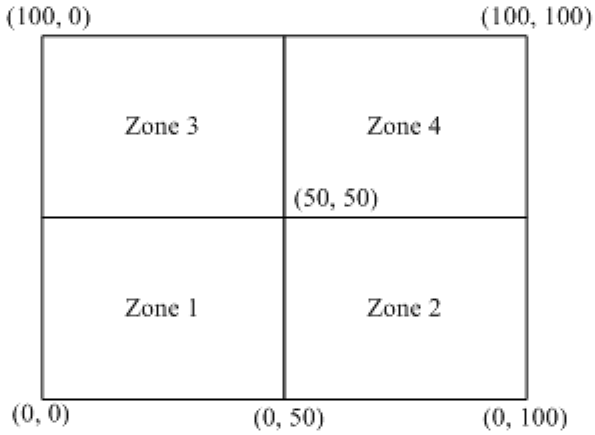

Figure 5.2: Zone creation

For instance, if the network area is very small (e.g., $100m \times 100m$) and divided into four zones, then sensor nodes are distributed as follows. Nodes with positions between (0, 0) and (50, 50) reside in Zone 1 whereas, sensor nodes with positions between (50, 0) and (100, 50) reside in Zone 2. Figure 5.2 illustrates the zones construction. We investigate the placement of BS either inside (Figure 5.3) or outside (Figure 5.4) of all zones in terms of network energy consumptions that is evaluated in Section 5.1.2. We find that placing BS at center of all zones and outside of all zones is same in terms network energy consumptions. Thus, we consider placing BS at outside of all zones.

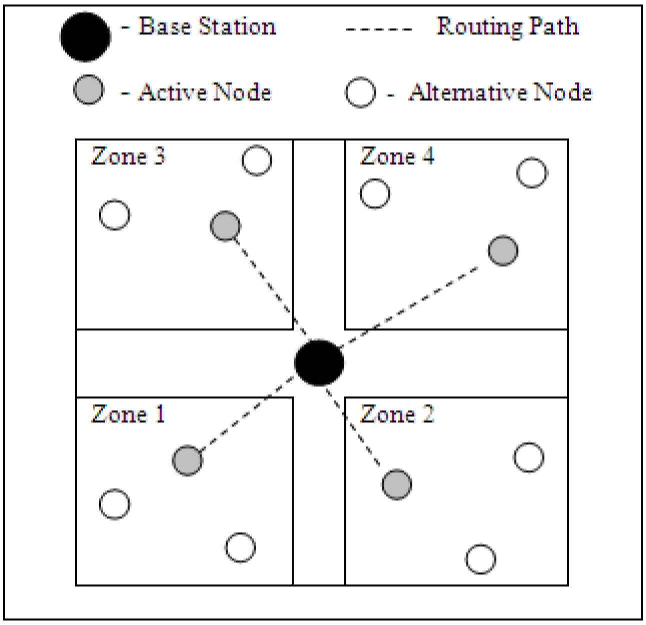

Figure 5.3: Zone-based topology with single-hop routing, BS is placed at center

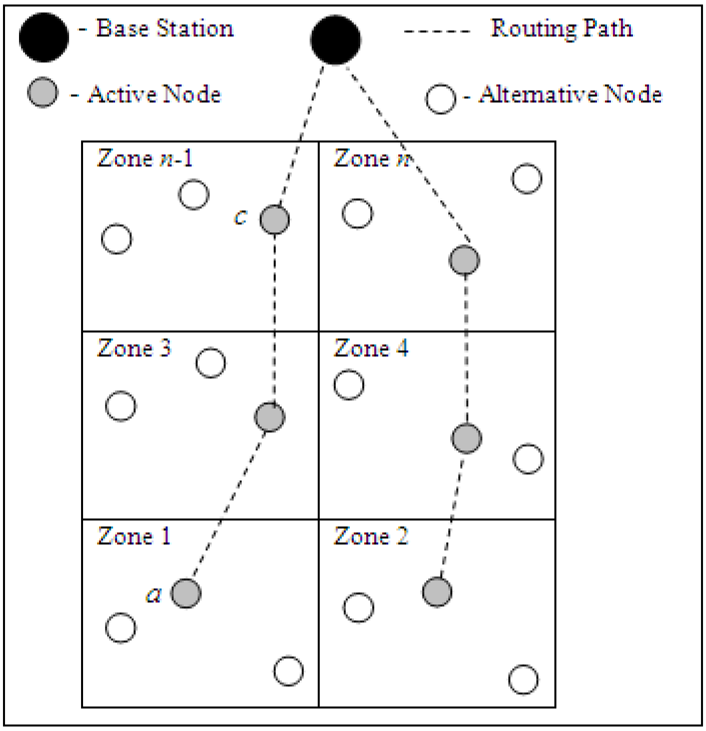

Figure 5.4: Zone-based topology using multi-hop routing, BS is placed outside of all zones

A few more sensor nodes are distributed to zones that are closest to BS than the zones that are farthest from BS. It is more likely that data collected by nodes of remote zones pass through the nodes of zones that are closest to BS. Hence, nodes in the zones that are closest to BS consume more energy and eventually have a shorter lifetime. Thus, we deploy a few more nodes in the zones that are closed to BS than the zones that are far from the BS. We present the experimental results of nodes distribution in Section 5.1.2, which shows that distributing a few more nodes to the zones that are closed to BS is more energy efficient than distributing equal number of nodes to all zones.

5.1.2 Experiment Results

We measure the effect of placing BS at center and outside of all zones and also distributing a few more nodes to the zones that are closest to BS. We run the simulation varying sensor's data collection time for a fixed number of zones and nodes. We use randomly connected Unit Disk Graphs (UDGs) on an area of 100 meters $\times$ 100 meters as a network simulation model. We set the number of zones and nodes to 6 and 36, place BS outside and center of all zones at the coordinates (50, 115) and (50, 53), respectively, and vary the sensor's data collection time between 3 and 12.

We run the simulation 25 times for each data collection time, and take the average of the 25 values as the remaining energy of all nodes of zones or networks for that particular data collection time. We run the same experiment by placing BS outside, and center of all zones at coordinates (50, 101) and (50, 51).

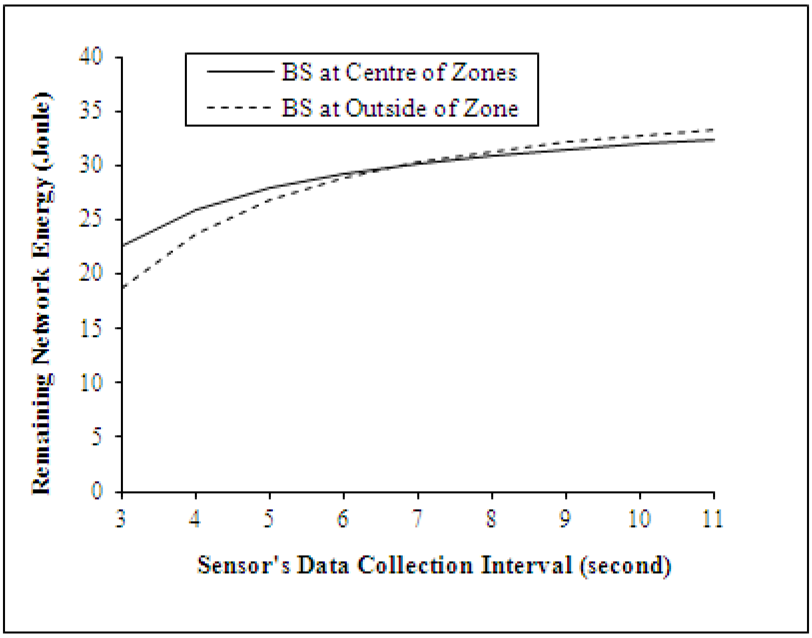

Figure 5.5: Remaining energy of network when BS is at the center location $(X, Y) = (50, 51)$ and outside location $(X, Y) = (50, 101)$

Base station (BS) at Center vs. Outside of all Zones

Figures 5.5 and 5.6 illustrate the remaining energy of network N_1 (BS is placed at center of all zones) and N_2 (BS is placed outside of all zones and a few alternative nodes are distributed to zones that are nearest to BS) at different data collection time of sensors. We observe that the remaining energy (in Joule) of networks N_1, and N_2 is very close to each other and overlaps at some data collection time. Hence, these systems are expected to be the same. We perform student's t-test which also shows that placing BS at center (N_1) and outside of all zones (N_2) have the same remaining energy or lifetime.

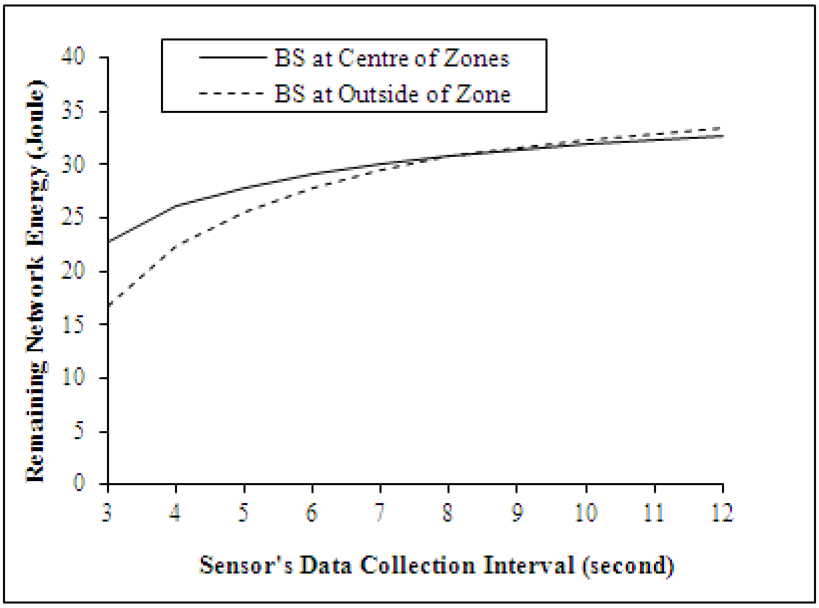

Figure 5.6: Remaining energy of network when BS is placed at the center location $(X, Y) = (50, 53)$ and outside location $(X, Y) = (50, 115)$

Equal number vs. a few Alternate Nodes distributed to Zones

Figures 5.7 and 5.8 present the remaining energy of networks at different data collection time of sensors, when BS is placed outside of all zones and (i) nodes are uniformly (equally) distributed to all zones and (ii) a few more nodes are distributed to zones that are nearest to BS than the zones that are farthest from the BS. We find that the remaining energy of network is more when a few more nodes are distributed to zones that are nearest to BS rather than distributing nodes uniformly to all zones.

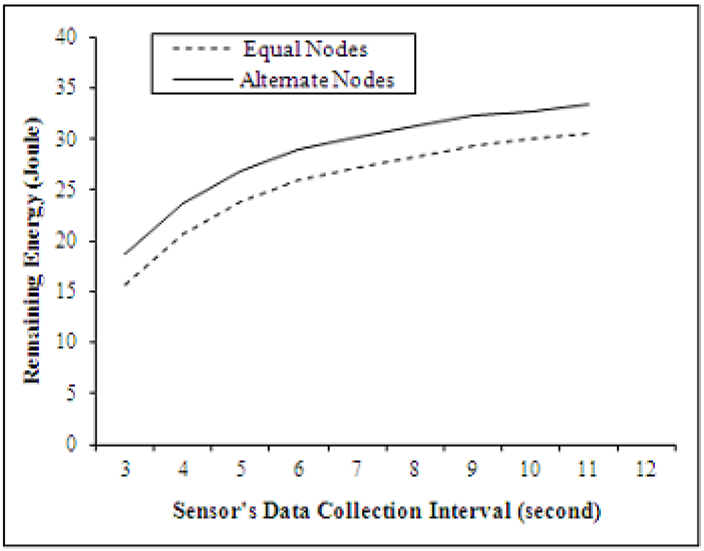

Figure 5.7: Remaining energy (lifetime) of networks that use Multi-hop routing, when BS is outside at (50, 101)

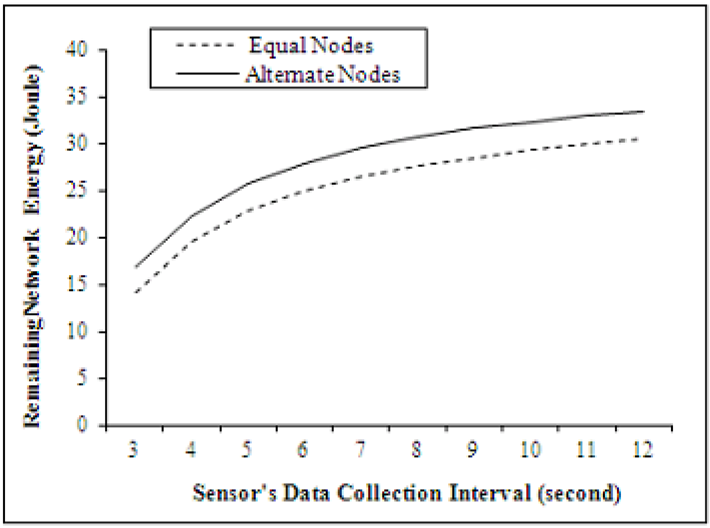

Figure 5.8: Remaining energy (lifetime) of networks that use Multi-hop routing, when BS is outside at (50, 115)

Similarly, we find that the zones (nearest to BS) with a few more nodes have more remaining energy than that of the zones with an equal number of nodes at different sensor's data collection time. Student's t-test also reveals the same phenomenon that zones that have a more nodes and are nearest to BS have a more lifetime than those with an equal number of nodes. In the following subsection we present the selection of active nodes by BS at each zone.

5.1.3 Active Node Selection

Each zone is divided into a number of squares where each square has one active sensor node. However, even if this assumption does not hold, the region of that square is sensed by an active node of the neighboring squares. This is because an active sensor node in a square has the sensing coverage of all the neighboring squares (Figure 5.9). Therefore, a very high probability of not having a sensing hole in the network is achieved. As a result, fault tolerance is accomplished. For instance, whenever the active sensor node of a square g_1 fails, the network operation (i.e., sending and receiving messages) proceeds. This is done without activating another sensor node for that square or re-establishing the network topology. This is because the active node of the neighboring square of g_1 fully covers g_1.

Using this topological structure, there is a high probability that no sensing hole exists in the network (Figure 5.9). The terms h, and R_s are the side of a square and the sensing range, respectively. Let us consider the following. The active sensor node a_1 of square g_1 exists at the bottom left corner and the active node a_2 of square g_2 exists at the top right corner. If the node a_2 fails, then the square g_2 can still be covered by the node a_1 of square g_1. This is true even if no other neighboring square of g_2 has active node. This is explained as follows. The farthest point p_2 of g_2 is within the sensing range of a_1. By following Pythagoras formula, the square coverage is possible when $R_s = 2\sqrt{2} \times h$. Figure 5.9 also clarifies this relationship between R_s, and h.

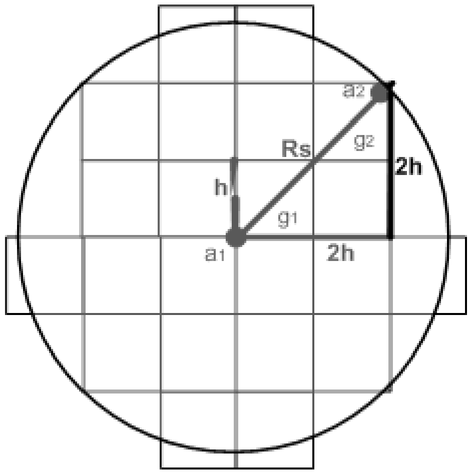

Figure 5.9: Zone-based topology when the sensing range, $R_s = 2\sqrt{2} \times h$

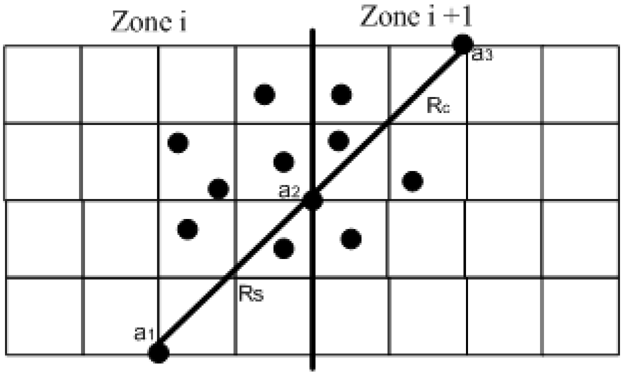

Figure 5.10: Zone-based topology with several active nodes within communication range of each other and $R_c = n \times R_s, n \geq 2$

This topological structure also ensures that several active sensor nodes (i.e., at least one sensor node) of a zone reside within the communication range (R_c) of the active nodes of the neighboring zones. This enables the sensor nodes to transmit the data to the BS through active nodes in their neighboring zones. This is because $R_c > R_s$ and there is a defined relationship between those ranges, i.e., $R_c = n \times R_s$ [113]. In Figure 5.10, we find that sensor node a_2 is at the border of two zones Z_i and Z_{i+1}. Sensor nodes a_1 and a_2 are at R_s apart. In addition, sensor nodes a_1 and a_3

are at R_c apart. In addition, a large number of nodes in Z_{i+1} reside within the R_c of node a_1 of Zone Z_i. Using this topological structure even if a small number of sensor nodes are distributed into the zones, the network is still expected to be covered. The reason is that each sensor node in a square covers all surrounding squares.

5.1.4 Cluster Head (CH) Selection

The BS chooses a Cluster Head (CH) or Zone Head (ZH) for each zone. CH, and ZH are interchangeably used in this paper. The chosen node is responsible for controlling the zone as well as selecting active nodes of a zone. The criterion for choosing the CH is as follows. Based on our assumption that all sensor nodes initially have the same residual energy, the BS randomly chooses a sensor node. The chosen node becomes the CH. If the residual energy of the sensor nodes varies, the sensor node with the highest residual energy is chosen as CH. Then, the CH chooses active nodes in each zone based on the following: 1) residual energy, 2) distance, 3) number of sleep rounds, and 4) free buffer spaces. If only one sensor node is located in a square, then it is chosen as an active node. Otherwise, having the same residual energy in nodes, say A and B in the same zone, node A is chosen as an active node. This is when node A is closer to the BS than B or active nodes of neighboring zones. Further ties are broken by comparing the remaining storage space or the number of sleep rounds of a node. Especially, node A is given more priority if it has more remaining storage space compared to node B. All other sensor nodes except the chosen active nodes remain in sleep-mode. This is done by turning their sensing, and transmission radios off. Thus, a number of active nodes is determined in each zone. The determined number of sensor nodes provides a high probability that these nodes provide the sensing coverage of the whole network. Then, each active node generates the shortest path to the BS. Once the CH and active nodes are chosen, they work for a certain period of time (or rounds). They stop working once their residual energy goes below a threshold value. Now, we proceed with the next component of the proposed framework (path establishment) as follows.

5.2 Path Establishment Method

The path establishment method works as follows. Sensor nodes at level L_1 calculate their distances to BS, and send the distance information to the sensor nodes at level L_2. Then, nodes at level L_2 calculate their distances to the sensor nodes at level L_1, and also the total distance to BS. Thus, active nodes at the level L_2 find out the shortest path to the BS (see Figure 5.11).

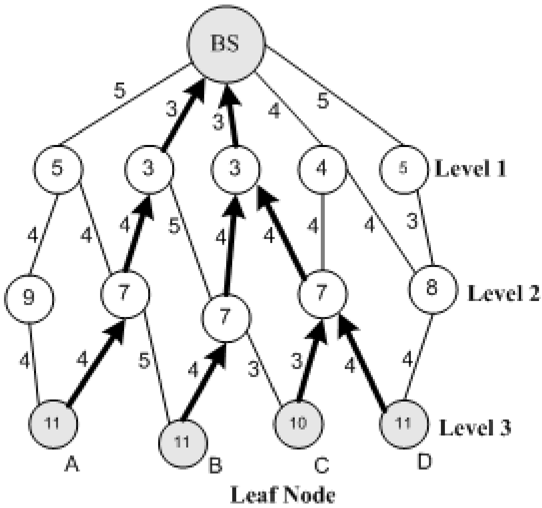

Figure 5.11: Shortest paths from leaf nodes to BS

Similarly, nodes at level L_k compute the total distance of the neighboring nodes at the upper level L_{k-1} to the BS. Afterwards, these nodes identify the shortest path to the BS. Based on this information, an active node creates a communication path with the neighboring active nodes. As a result, the total distance to the BS as well as the power consumptions is minimized as compared to existing tree based routing protocols since power consumptions are proportional to the distance of transmission path.

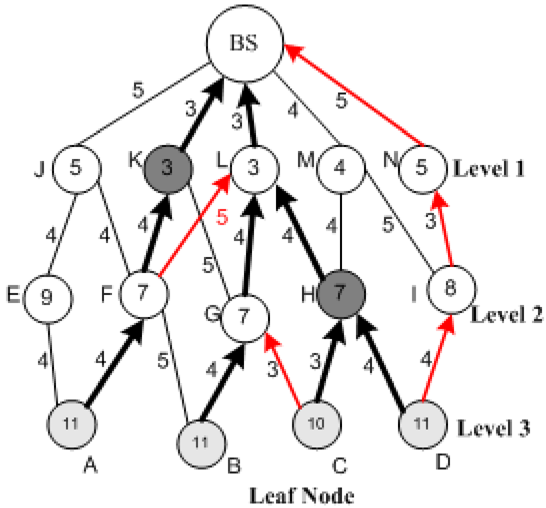

Figure 5.12: Alternative paths whenever a node fails

Moreover, each active sensor node x at the level L_i chooses another active sensor node y at the level L_{i-1} with which x has the second lowest distance to the BS. The sensor node y acts as an alternative node or as an intermediate node of an alternative path for each leaf node to the BS. This enables the leaf node to transmit data to the BS in case of any active node fails. Once the paths are established, active nodes sense, and route the data to the BS. Figure 5.11 demonstrates the path establishment method, where the shortest path from a leaf node to the BS is represented by bold arrows. Figure 5.12 illustrates the alternative path establishment, and the transmission of data through alternative node when an active sensor node fails. For instance, if node K at level 1 fails, active sensor node F at level 2 transmits data through node L. The alternative path is shown as red arrows. Based on the proposed framework, a new set of graphs for underlying topologies is proposed as follows.

5.3 A Set of Topology Control Algorithms

Recent research [21] have significantly considered studying the actual structure of WSN through graph theory. In particular, geometric graphs are used in WSNs [57] to model the relationship between a sensor node, and its neighboring sensor nodes. The Gabriel Graph (GG) [81, 29] uses a forbidden area to define the edges created between a set of sensor nodes. The constructed edges are undirected. In addition, the GG is a sub-graph of the UDG, where the UDG is connected. The algorithm for constructing the GG is as follows (see Figure 5.13). An edge binding two vertices, say v_1 (i.e., running the algorithm), and v_2 is in GG if the following criterion holds: the minimal diameter circle that circumscribes v_1 and v_2 whose diameter is the line segment that has both v_1 and v_2 as its endpoints, has no other vertex of V. The sensor node v_1 follows these steps with the other neighbors. Moreover, the algorithm is executed on every sensor node in the network.

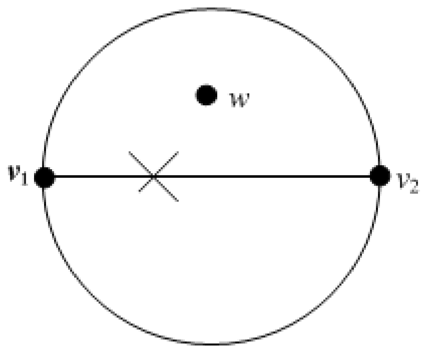

Figure 5.13: An edge $[v_1, v_2]$ is not in GG

Then, we introduce a new set of graphs referred to as the Mini Gabriel (MG) geometric graphs. These graphs are sub graphs of the Unit Disk Graphs (UDGs). Each sensor node, u, running these graphs chooses the nearest neighboring node, v. Afterwards, a midpoint *mid* is calculated between the sensor nodes u and v. As a result, a circle c is constructed with a center located at *mid* with a radius *r_mid* equal to $s \times (|uv|/2)$. The term s is a constant parameter within the range of [0, 1], which shortens or increases the r_mid. If c contains a sensor node, say w, in its area, then the edge $[u, v]$ is removed. Otherwise, the edge $[u, v]$ is preserved.

The algorithm proceeds on each sensor node until the network originally modeled by the UDG is reconstructed. This construction is based on a given s value. Because the MG only needs the position of the sensor node running this algorithm (i.e., current sensor node) as well as its neighboring nodes, this algorithm is therefore local. If s is equal to one, then a special case of the graph is constructed, a Gabriel Graph (GG) [81, 29]. However, the new set of graphs preserves more edges by using smaller circles. This is compared to that circle constructed by the GG. The possibility of maintaining shorter paths is therefore increased in the MG. This is compared to the GG, which makes the MG a denser graph. The MG runs in $O(l * l)$, where l is the degree of sensor nodes. This is because each time a sensor node, say u, runs the MG, chooses the nearest neighbor. After words, u checks all the possible neighbors if they are inside the circle constructed. In addition, the algorithm behaves locally which means that the complexity is analyzed per one sensor node. For completeness, the MG's algorithm is included as follows (see Algorithm 1).

In the following subsection, we present the framework's component, routing sensors data.

5.4 Routing of Sensors Data

The proposed routing protocol has two phases: (1) setup and (2) steady phase. The setup phase includes zones creation, node distribution active nodes selection, and path establishment. All the framework's components are based on MG and GG. In the steady phase, data is transmitted from the source active node to the BS through intermediate active nodes. Active nodes at different levels work using TDMA scheme.

Algorithm 1 $MG(G, s)$ graph algorithm

Input: A graph G with the sensor node set V and a parameter s.

Output: A list of undirected edges L for each sensor node $u \in V$, which represents the MG subgraph of G, $MG(G)$

for all node $u \in V$ **do**

Create a list of neighbors of u: $LN(u) = N(u).L = \phi$

end for

while $LN(u) \neq \textit{empty}$ **do**

Remove the nearest neighbor sensor node v from $LN(u)$

Scan the list $LN(u)$ and add the undirected edge $[u, v]$ to L if no node w exists that is inside the circle c with a center $= \frac{u+v}{2}$ and a radius $= s.(|uv|/2)$

end while

The length of the timeslots at different levels is variable. For instance, at Timeslot 1, the active nodes at the farthest level L_k sense the subscribed events. Furthermore, the nodes send the corresponding data to active nodes at upper level L_{k-1}. At Timeslot 2, active nodes at the level L_{k-1} receive data packet. These nodes send acknowledgements to nodes at the level L_k. Moreover, they transmit data packet to nodes at the upper level L_{k-2}. Hence, the length of Timeslot 1 is shorter than that of Timeslot 2.

Setup phase is initiated at a certain number of rounds, R_n, which also depends on the energy status of sensor nodes, and network. Each node keeps track of the number of message transmitted and sends this information along with the data packet to BS. BS estimates the total network energy consumptions of each node and network. Thus, the network energy consumptions is properly distributed. Data transmission from the source node to the BS is completed at each round. where, number of rounds, R_n is calculated using the Equation 5.1.

$$R_n = \frac{PrevNumOfRounds}{PrevNetEnergy} \times CurrentNetEnergy \tag{5.1}$$

The proposed $CRTCA$ protocol achieves fault tolerance, which is presented in the following section.

5.5 Fault Tolerance

In the proposed $CRTCA$ protocol, there are alternative paths for any established path between a source node and BS. Thus, the proposed protocol achieves fault tolerance, which represents the continuous operation of network. It has two phases: i) detecting failure of nodes (Fault Detection) ii) then route data through alternative paths (Fault Recovery).

Fault Detection

BS subscribes to nodes for some events of interests, such as "temperature greater than 40 degree Celsius". If an active node, x of the current data transmission path senses such event of interest at its allocated timeslot, x sends the data packet to the active node at upper level. If x has no subscribed event to send to the active node at upper level, x sends a small sized special packet to notify the active node at the upper level that it (x) is still alive. If neighboring active node or BS does not receive any data or special packet from an active node BS assumes that the active node has failed. Then, BS excludes the node from its allocated timeslot and assigns alternative node/path to transmit the data.

Fault Recovery

Fault recovery is achieved by allocating alternative paths for any established path between a source node and BS. Hence, data is retransmitted through alternative paths if a node failure is detected. This has been presented in Section 5.2 in detail (Figure 5.12).

5.6 Performance Evaluation: CRTCA, CRTCA-MG, and CRTCA-GG

In this section, we present the energy model, the performance metrics, and the simulation model we have used for the performance evaluation of the proposed Cluster-based Routing and Topology Control Approach (CRTCA) and its variant with Gabriel (GG), and Mini Gabriel (MG) graphs. Furthermore, we show the simulation results with performance comparison.

5.6.1 Energy Model

The energy consumption of a sensor node for sending a data packet of size n_{data} bytes to another sensor node at the distance d is

$$E_{Tx} = n_{data} \times \epsilon_{data} + n_{data} \times d^2 \times \epsilon_{air} \tag{5.2}$$

$$E_{Rx} = n_{data} \times \epsilon_{data} \tag{5.3}$$

In Equation 5.2, ϵ_{data} represents energy spent in transmitter electronics circuitry (50 $nenoJoule/bit$) and in Equation 5.3, ϵ_{air} represents energy consumptions in radio amplifier for propagation loss (0.01 $nJ/bit/m^2$). Receiving energy consumptions of a sensor does not have energy consumptions of radio amplifier for propagation loss.

5.6.2 Simulation Model and Performance Metrics

We simulate the performance of 1) Cluster-based Routing and Topology Control Approach (CRTCA) framework, 2) CRTCA based on the Mini Gabriel graph (CRTCA-MG), and 3) CRTCA based on the Gabriel Graph (CRTCA-GG) using the C programming language. We use randomly connected Unit Disk Graphs (UDGs) on an area of 100 meter $\times$ 100 meter as a basis of our simulation model. Base station (BS) divides the network into a number zone and CH selects a number of active nodes in each zone that establish path to BS. The co-ordinate of BSis assumed to be at (55,101).

The reasons for choosing GG in our comparisons are as follows. First, MG set's elimination areas are derived from GG's elimination area. Second, GG has been generally used as underlying topologies for WSNs [73]. Third, GG reduces the degree of nodes but does not remove too many edges compared to other existing geometric graphs [21, 103]. Therefore, GG preserves the shortest, and least power paths compared to those existing graphs.

The evaluation is done in terms of the following: a) network energy consumptions (i.e., transmission and receiving power), b) transmission energy consumptions, c) end-to-end data transmission delay, d) accumulated routing delays, and e) length of the paths in terms of the crossed hops. Network energy consumption is defined as the total energy consumed by all the sensors nodes for routing data over a certain period of time. Network energy consumptions also reflect the lifetime of the network, i.e., the remaining network energy. We measure the end-to-end delay as the time that is required to transmit data from any source sensor node to the BS based on the traversed Euclidean distances. Another type of delay is in terms of the routing delay. Using this metric, the accumulated routing delays from any source sensor node to the BS are measured. Between these pair of nodes, we also use another metric that counts the number of hops traversed on the path way which actually measures the network's paths lengths.

Some simulation runs measure these metrics by either varying the followings: 1) the number of rounds (num_r) (i.e., from 10000 to 40000 of an increment of 10000) and 2) the number of zones (num_z) (i.e., from 4 to 10 of an increment of 2). When varying num_r, we fix num_z and the number of sensor nodes (num_sens), to 4 and 120, respectively. However, when we vary the num_z, we fix the num_r and the num_sens to 10000 and 120, respectively. For all these simulation runs, the constant s for the Mini Gabriel (MG) graph is varied from 0.25 to 0.75 with an increment of 0.25.

5.6.3 Simulation Results

Figures 5.14 and 5.15 illustrate the total network energy consumptions varying the number of zones in the network and number of rounds, respectively. In both cases, the CRTCA outperforms other approaches. The same observation applies even if s is varied for the MG. This behavior is due to the high number of intermediate sensor nodes along the source-destination paths constructed by the CRTCA-GG and the CRTCA-MG. In other words, increasing the values of s also increases the radius for the circular forbidden area where the intermediate nodes reside. Thus, more sensor nodes expose their battery power. As a consequence, the total network energy consumptions for the CRTCA-GG and the CRTCA-MG are more than that for the CRTCA. It is also evident that if there are more intermediate nodes on a path in GG and MG, the receptions energy consumptions at the intermediate nodes will contribute to the larger network energy consumptions.

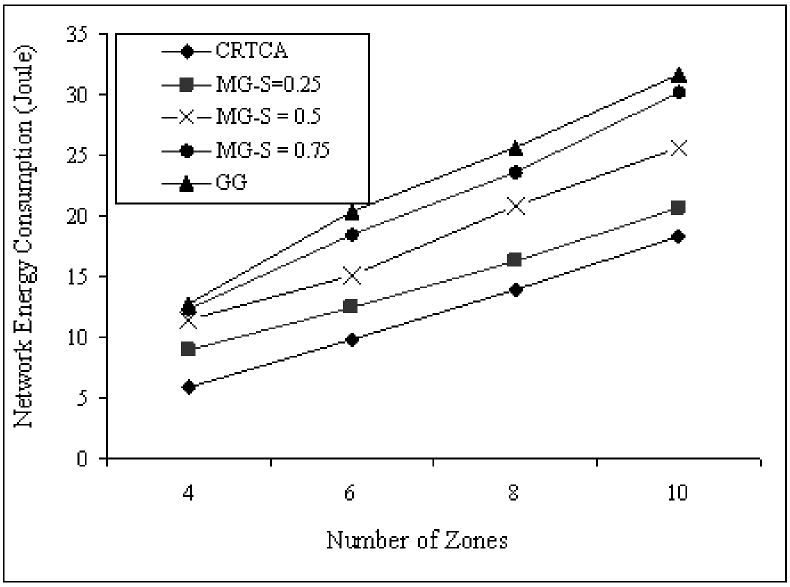

Figure 5.14: Network energy consumptions varying number of zones,num_z

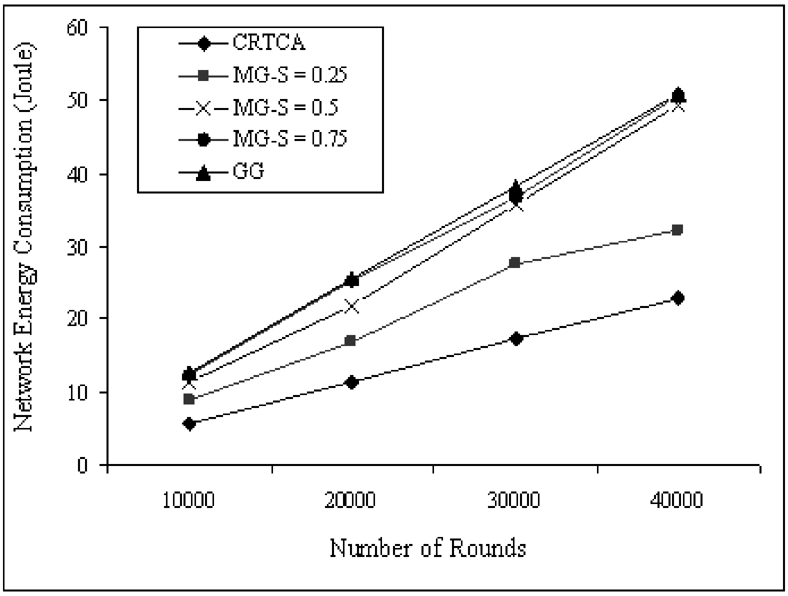

Figure 5.15: Network energy consumptions varying number of rounds, num_r

Hence, we evaluate their performance in terms of only the transmission energy consumptions which are illustrated in Figures 5.16 and 5.17. Again, we find that CRTCA outperforms the CRTCA-GG and CRTCA-MG. This is because more number of transmissions through the intermediate nodes will have more overhead energy consumptions of radio amplifier for propagation loss.

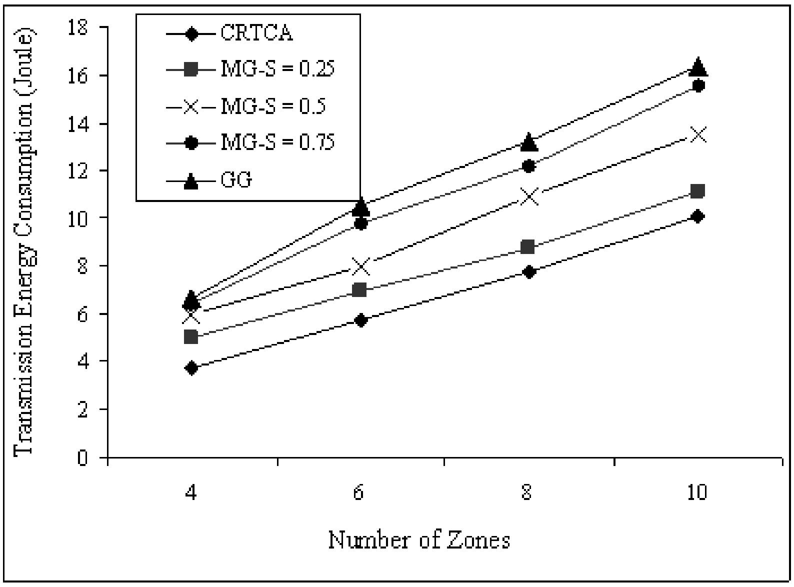

Figure 5.16: Transmission energy consumption varying number of zones, *num_z*

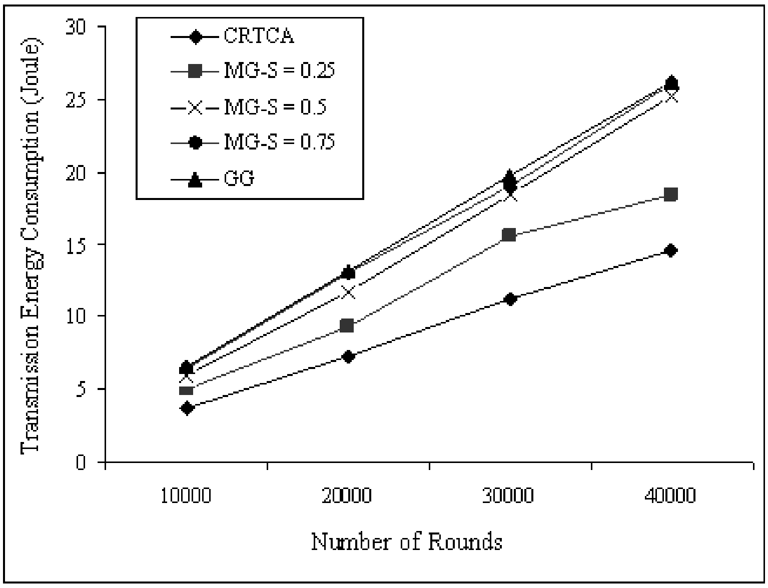

Figure 5.17: Network energy consumptions varying number of rounds, *num_r*

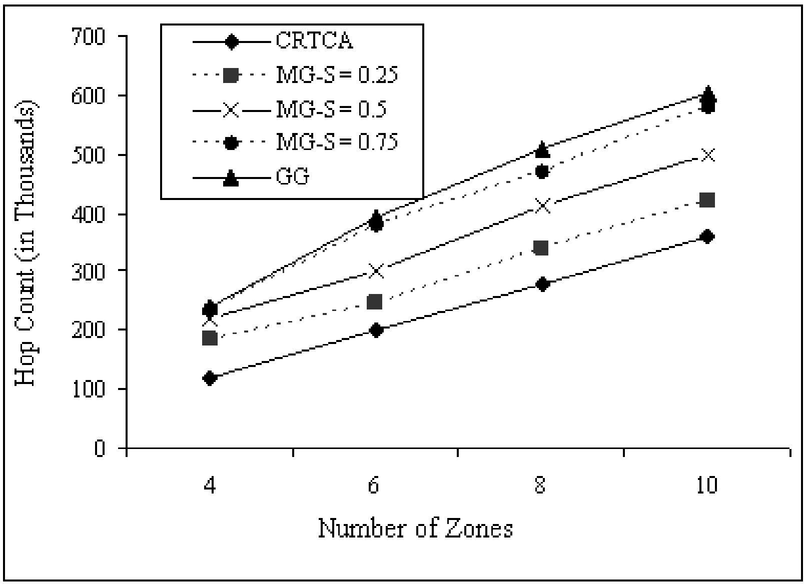

Figure 5.18: Paths lengths in terms of hops varying number of zones, num_z

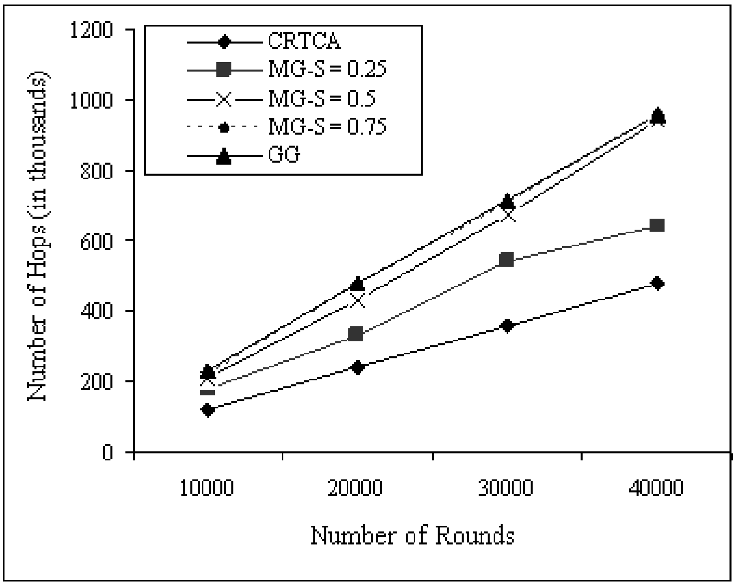

Figure 5.19: Paths lengths in terms of hops varying number of rounds, num_r

Figure 5.18 and 5.19 support the previous explanation by showing that the $CRTCA$ maintains more number of shorter paths in terms of hops compared to those paths traversed by the other approaches.

From all the previous simulation runs, an interesting behavior for the set of MGs is demonstrated when s is varied. The behavior is as follows: when s increases the possibility for loosing edges increases as the radius for the circular forbidden area increases. As a consequence, more intermediate nodes are selected for transmitting data from a source node to the destinations that results more network energy consumptions, transmission energy consumptions, and number of hops.

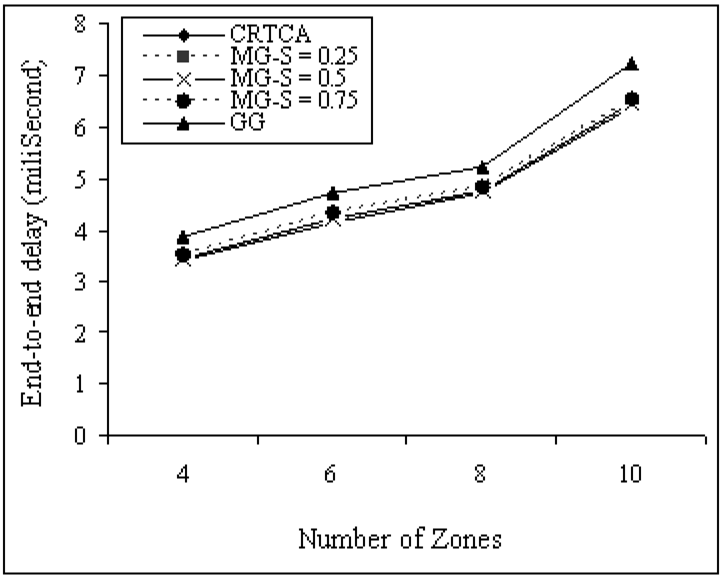

Figure 5.20: End-to-end delay varying number of zones, num_z

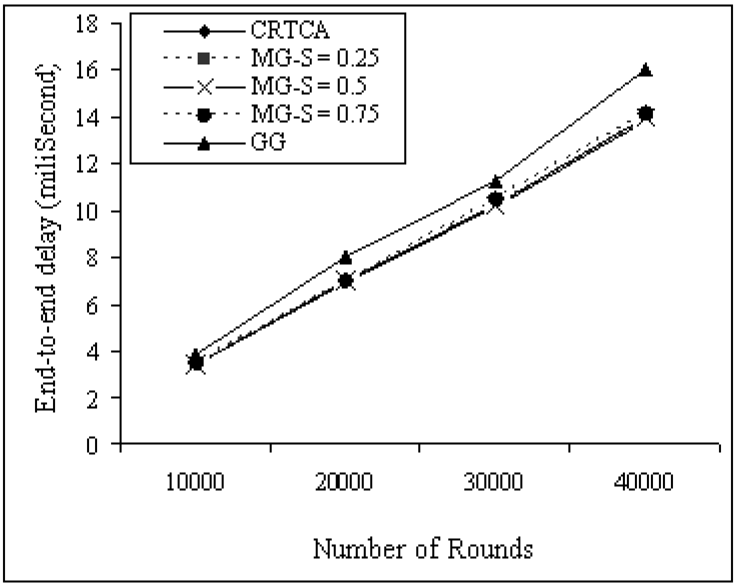

Figure 5.21: End-to-end delay varying number of rounds, num_r

Figures 5.20 and 5.21demonstrate that the CRTCA and CRTCA-MG have almost similar performance but CRTCA is better than CRTCA-GG in terms of end-to-end delay both for varying the number of zones and rounds. One of the possible reasons is that the CRTCA keeps more edges compared to those for the GG. As a result, shorter paths are discovered by the CRTCA, which significantly improves the transmission delay. However, we found significant differences among CRTCA, CRTCA-MG and CRTCA-GG using ANOVA test at 95% confidence level since the value of F-calculated (5.7972) is greater than the value of F-table (2.525).

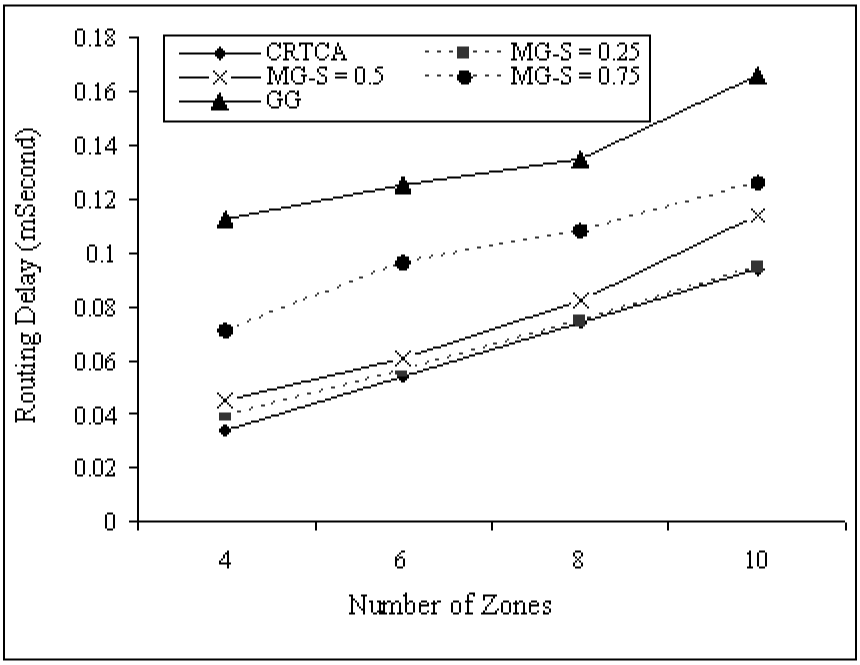

Figure 5.22: Routing delay varying number of zones, num_z

Finally, Figure 5.22 demonstrates the routing delay for varying the number of zones. Again, CRTCA outperforms both MG and GG since in CRTCA, a path has less number of intermediate nodes and thus, delay for making routing decision will be less as compared to MG and GG, which have more intermediate nodes on a data path.

In general, the above results demonstrate that CRTCA outperforms MG and GG in most cases. For instance, as the number of zones increases the number of levels in the hierarchy also increases. As a consequence, the number of sensor nodes is decreased in each subzone of a level. Therefore, the average degree of sensor nodes is decreased. Furthermore, because of the nature for the MG and the GG, more edges are removed. Thus, longer paths are achieved between any pair of sensor nodes in the GG and the MG that contribute to more network energy consumptions, transmission energy consumptions, and end-to-end delay. We also perform student's t-test at 95%

confidence level and in most cases we find that the p-value is smaller than 0.05, which reveals the same phenomenon, as are demonstrated and stated. However, all these simulations demonstrate that the MG achieves the connectivity property.

5.6.3.1 *Individual Nodes Energy Dissipation*

We also evaluate the performance of the proposed routing protocol with topology N_2 (a few more nodes are distributed to zones that are closed to the BS) against topology N_1 (equal number of nodes are distributed to all zones) in terms of the remaining energy of individual sensor nodes for several energy groups (in percentage). This experiment is performed to illustrate whether the energy dissipation of individual sensor nodes is properly distributed.

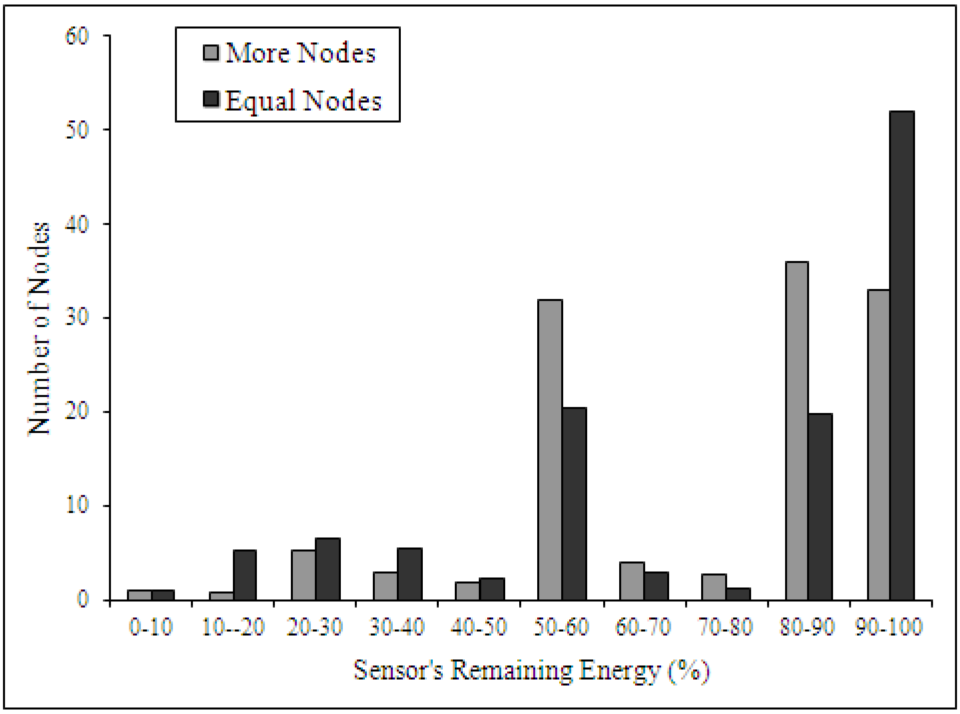

Figure 5.23: Remaining energy of sensor nodes (in percentage) for equal nodes and extra nodes distributed to zones that are closed to BS

Let us assume that S_n and Z_n are the number of sensors, and zones, respectively, in the network, p is the number of more nodes that are distributed to the zones closest to BS. If S_n is divisible by Z_n, $\frac{S_n}{Z_n}$ nodes are distributed to each zone of N_1. However, the number of nodes to the zones that are closed to BS of N_2 will be $\frac{S_n}{Z_n} + p$. For instance, in this simulation, we set the number of nodes, and zones to 120, and 4 respectively. Thus, 30 nodes are distributed to each zone of N_1, whereas, the number of nodes in the zones are 36, 36, 24, 24 for N_2 since $p = 6$.

To evaluate the performance we divide the sensor's remaining energy into 10 percentage groups: $0 - 10\%, 11 - 20\%, \ldots, 90 - 100\%$. We run the simulation by varying the number of rounds and take the average of remaining energy in different energy groups. Figure 5.23 illustrates that the number of nodes that lie in the lowest percentage groups (in the range $0 - 50\%$) are less in N_2 (i.e., More Nodes) than in N_1 (i.e., Equal Nodes). Thus, N_2 has better performance than N_1 since it is highly expected to have less number of nodes in the lower percentage groups. This also reflects the network lifetime, and energy dissipation of individual nodes of the network.

To better understand the energy dissipation of individual sensor nodes we also divide the network remaining energy in four major percentage groups: $0-29\%, 30-59\%, 60-89\%, 90-100\%$. Then, we vary p from 4 to 10 for a fixed number of rounds and take the average of remaining energy in four groups for a number of simulation runs. Figure 5.24 illustrates that the remaining energy of sensor nodes are properly distributed in each energy group. This also reflects the prolonged network lifetime since the number of nodes in the lowest remaining energy group ($0 - 29\%$) is much less than the other groups. Again, the lifetime of N_2, "More Node" will be longer than that of N_1,"Equal Nodes" since the number of nodes of N_2 in the energy group $0 - 29\%$ is much less than that of N_1. Similarly, Figure 5.25 illustrates the number of nodes that lie in different energy groups for varying p between 4, and 10.

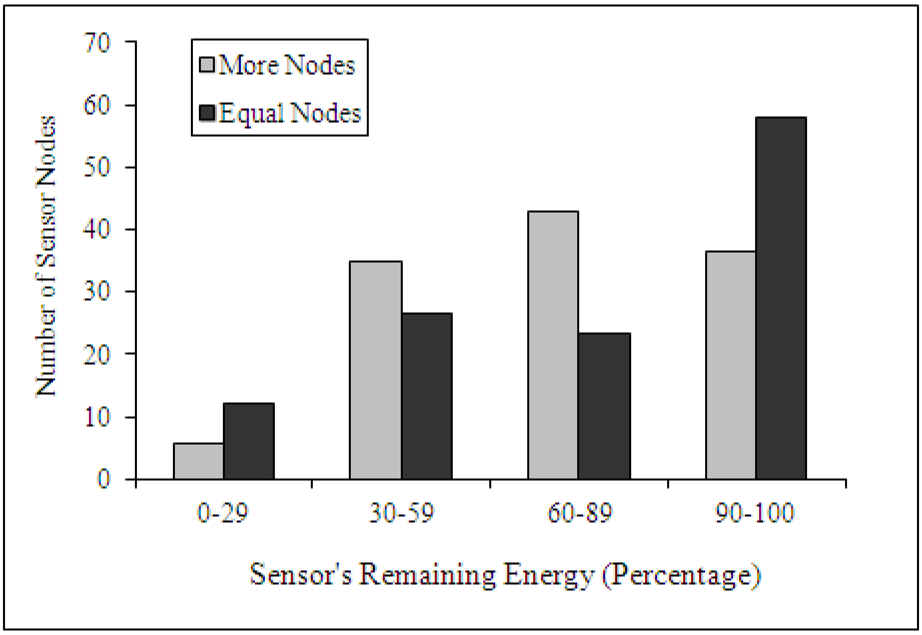

Figure 5.24: Remaining energy of sensor nodes (in percentage) for equal number of nodes and extra nodes distributed to zones that are closed to *BS*

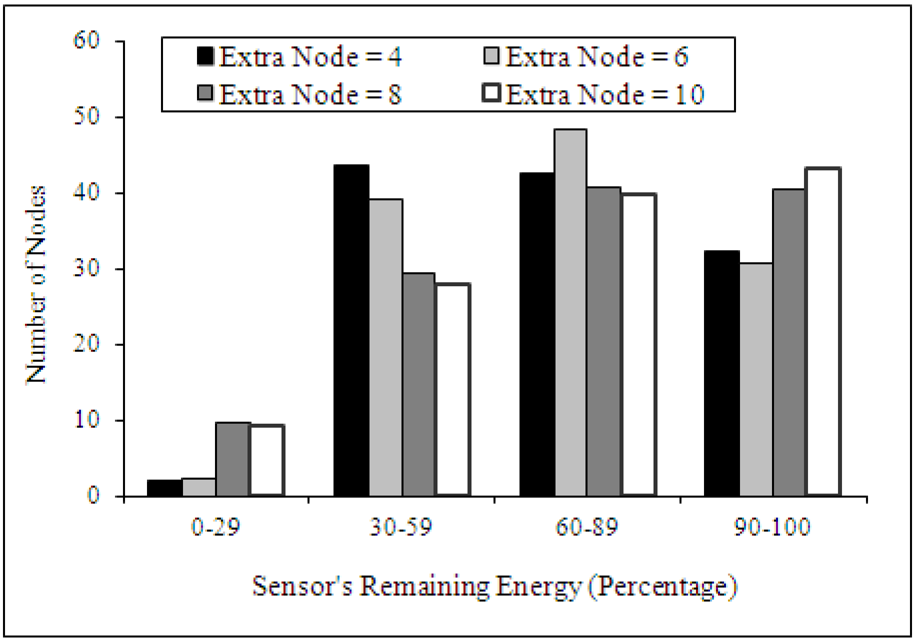

Figure 5.25: Remaining energy of sensor nodes (in percentage) varying the number of extra nodes that are distributed to the zones that are closed to *BS*

Thus, we can conclude that the proposed routing protocol, CRTCA balances individual nodes energy consumptions as well as the overall network energy consumptions. A few of the possible reasons are: (i) a few more nodes are distributed to zones that are closed to BS since nodes in these zones receive, and transmit more data packets as compared to nodes in other zones, (ii) each node sends its remaining energy information with the data packet that is transmitted to BS; thus, BS can reselect active nodes if the remaining energy of an current active node goes below a threshold value, (iii) after a certain number of rounds, R_n the setup phase is initiated, where R_n is calculated based on the current network energy (Equation 5.1).

5.6.3.2 Data Packet Delivery Ratio

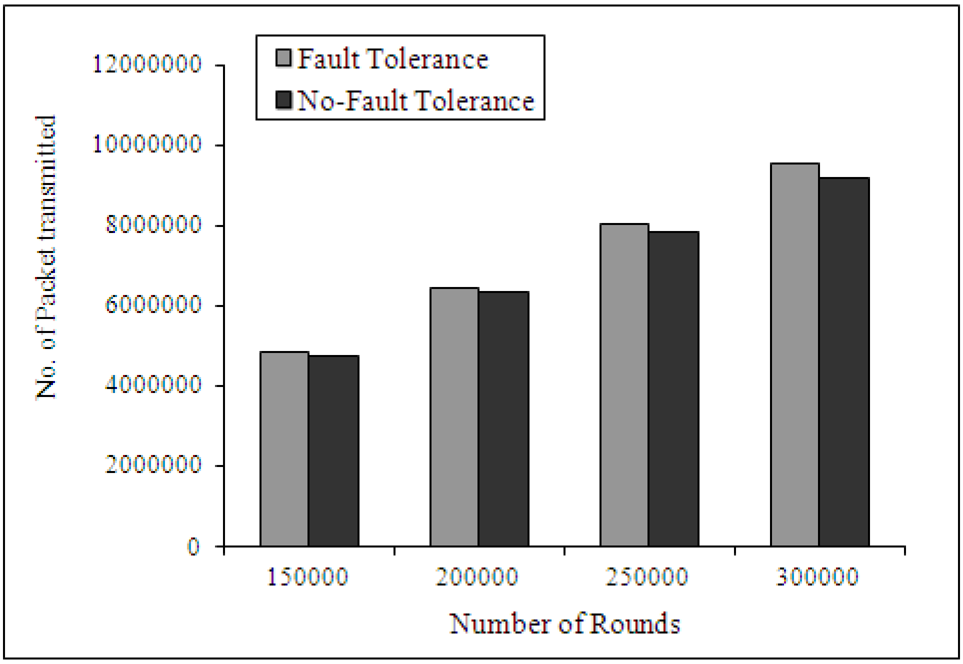

Figure 5.26: Number of packets transmitted using fault tolerance and without fault tolerance technique varying the number of rounds

Figures 5.26 and 5.27 illustrate the fault tolerance characteristics of the proposed routing protocol in terms of number of transmitted packets. Figure 5.26 shows that the number of transmitted packets for No-Fault Tolerance (i.e., no fault tolerance mechanism is implemented) is less than that with Fault Tolerance (i.e., a fault tolerance mechanism is implemented). Thus, a number of packets are lost due to the links and/or nodes failures whenever no fault tolerance mechanism is implemented in the proposed routing protocol.

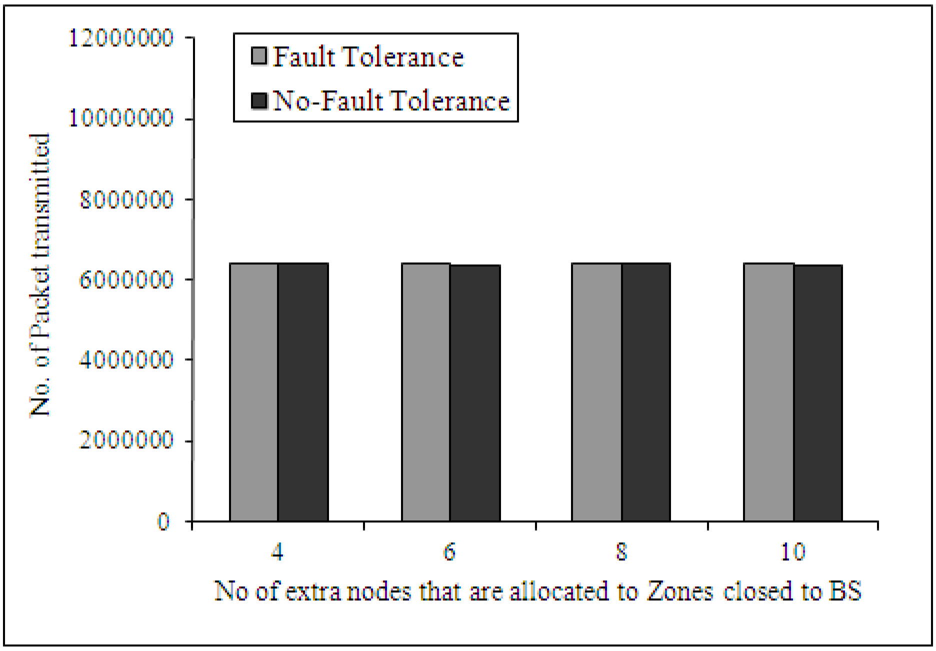

Figure 5.27: Number of packets transmitted using fault tolerance and without fault tolerance technique for equal nodes and extra nodes distributed to zones that are closed to BS

Figure 5.27 illustrates that only a small number of packets are lost for the number of extra nodes, $p = 6$ and $p = 10$, whenever no fault tolerance mechanism is implemented in the protocol. This result reflects that there is a probability of packet loss without taking any measures for fault tolerance.

5.7 Performance Evaluation: CRTCA, LEACH, and DSC

In this section, we evaluate the performance of the proposed CRTCA protocol [52, 53], and compare with the existing DSC [8], and LEACH [38] protocols through mathematical analysis, and simulation.

5.7.1 Analysis

Using the energy model that is presented in Equations 5.2 and 5.3, we calculate the energy consumptions of a node for sending a data packet of size n_{data} bits to the BS at the distance d as

$$ETx_{data}(n_{data}, d) = n_{data} \times \epsilon_{elec} + n_{data} \times d^2 \times \epsilon_{air} \tag{5.4}$$

Energy consumptions of a node for sending a special packet of size n_{spec} bits to the BS at the distance d are

$$ETx_{spec}(n_{spec}, d) = n_{spec} \times \epsilon_{elec} + n_{spec} \times d^2 \times \epsilon_{air} \tag{5.5}$$

We assume that the probabilities of a node transmitting data packet and special packet are P_{data} and P_{spec}, respectively. Hence, the total energy consumptions of a node considering that each node sends either the data packet (only when the subscribed event occurs) or a special packet (to notify that the node is still alive) to the BS are

$$E_{Tx} = P_{data} \times ETx_{data}(n_{data}, d) + P_{spec} \times ETx_{spect}(n_{spec}, d) \tag{5.6}$$

Similarly, the energy consumptions of a sensor node due to the receipt of a data or special packet is

$$E_{Rx} = P_{data} \times n_{data} \times \epsilon_{elec} + P_{spec} \times n_{spec} \times \epsilon_{elec} \tag{5.7}$$

As the size of the special packet, n_{spec} is much smaller than that of the data packet, n_{data}, we assume that

$$n_{data} = k \times n_{spec} \tag{5.8}$$

where, k is a constant. In simulation, we assume the value of k is at least 32.

Moreover, the probability of sending data packets, P_{data} and special packets, P_{spec} has the following relationship.

$$P_{data} = l \times P_{spec} \tag{5.9}$$

Where $0 < l \leq 1$. Now, by placing the value of n_{spec} of Equations 5.8 and value of P_{spec} of Equation 5.9 into Equation 5.6 we get the following Equation 5.10.

$$\begin{aligned} E_{Tx} &= P_{data} \times ETx_{data}(n_{data}, d) + P_{spec} \times ETx_{spec}(n_{spec}, d) \\ &= P_{data} \times ETx_{data}(n_{data}, d) + \frac{P_{data}}{l} \times \frac{1}{k} \times ETx_{data}(n_{data}, d) \\ &= ETx_{data}(n_{data}, d)(P_{data} + \frac{P_{data}}{lk}) \end{aligned} \tag{5.10}$$

From Equation 5.10 the transmission energy consumptions of a node in the DSC protocol can be calculated and related to that in the CRTCA as follows.

$$\begin{aligned} ETx_{data}(n_{data}, d) &= \frac{E_{Tx}}{(P_{data} + \frac{P_{data}}{lk})} \\ &= m \times E_{Tx} \\ \Rightarrow E_{DSC-Tx} &= m \times E_{CRTCA-Tx} \end{aligned} \tag{5.11}$$

In Equation 5.11 E_{DSC-Tx} and $E_{CRTCA-Tx}$ represent the energy consumptions of a node for transmitting data in the DSC and CRTCA protocols, respectively and

$$m = \frac{1}{(P_{data} + \frac{P_{data}}{lk})} \tag{5.12}$$

Considering the values of k and l in Equations 5.8 and 5.9 respectively, the value of the denominator of Equation 5.12 will be always less than 1. Thus, the value of m in Equation 5.12 will be $m > 1$. For instance, if $P_{data} = 0.9$ then $P_{spec} = 0.1$ and $l = 0.9/0.1 = 9$ (Using Equation 5.9). Using Equation 5.11 and $k = 16$ we find $m = \frac{1}{0.9+\frac{0.9}{9\times 16}} = 1.1034 > 1$. Similarly, for any value of P_{data}, and P_{spec}, $m > 1$.

Hence, Equation 5.11 implies that a node consumes more energy for data transmissions in the DSC protocol than that in the CRTCA protocol.

Similarly, we can relate the DSC protocol with the CRTCA protocol in terms of the receiving energy consumption. From Equation 5.7, we can easily find the following relationship.

$$E_{DSC-Rx} = m \times E_{CRTCA-Rx} \tag{5.13}$$

Where E_{DSC-Rx} and $E_{CRTCA-Rx}$ represent the energy consumptions of a sensor node for receiving data in the DSC and CRTCA protocols, respectively. From this analysis, it can be concluded that our proposed CRTCA protocol is more energy efficient than that of the existing DSC protocol.

Similarly, we can show that CRTCA is more energy efficient than LEACH because a setup phase is initiated at the end of each round in LEACH whereas, after a certain number of rounds a setup is initiated in CRTCA. Moreover, in the CRTCA protocol, a node sends a special packet to notify that it is still alive. This provides fault tolerance in the CRTCA protocol.

5.7.2 Simulation Setup and Results

We simulate the proposed $CRTCA$ protocol using the energy model that is presented in Section 5.7.

Table 5.1: Simulation parameters, and their respective values

Parameter	**Value**
Network Size	100m X 100m
Number of Nodes	Maximum 100
Number of Zones	Maximum 8
Base station position	(90, 170)
Data packet size	256 bits
Transmission Energy Consumptions	50 $nJoule/bit$
Energy Consumption in free space or air	0.01 $nJoule/bit/m^2$
Initial Node Energy	2 Joule

Based on the parameters, and their respective values that are presented in the Table 5.1, we perform the simulation of the LEACH, DSC, and CRTCA protocols and measure their performance in terms of energy consumptions, number of transmission, and network lifetime (or energy dissipation).

Simulation results show the energy consumptions of LEACH, and DSC protocols are much more than that of the CRTCA protocol (Figure 5.28). The energy consumptions include energy consumptions of a sensor node for transmitting data and special packets to BS, and for receiving data and special packets, aggregating data packets, and sending to BS. This is because the CRTCA protocol transmits small sized special packets, which consume less energy. Moreover, the CRTCA protocol initiates a setup phase after a certain number of rounds, where the number of rounds is dynamically calculated based on the network energy status. ANOVA test at 95% confidence level also reveals the same phenomenon that energy consumptions of sensor nodes over a number of rounds are much more in LEACH and DSC protocols than that in the CRTCA protocol since the value of F-calculated (24.46) is much higher than the value of F-table (3.9).

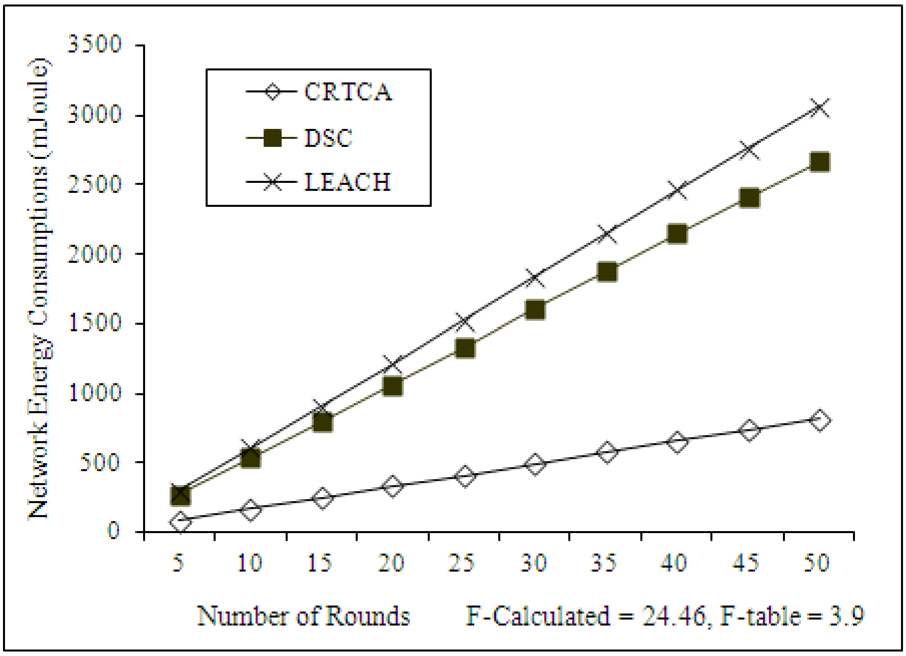

Figure 5.28: Network energy consumptions of CRTCA protocol as compared to DSC and LEACH protocols

Figure 5.29 illustrates the number of communications that takes place over a number of rounds. This includes data and special packets transmission, message transmission for selecting active nodes in CRTCA protocol, message transmissions for new CHs selection and cluster formation. It is observed that the communication overhead is higher in the LEACH and DSC protocols than that in the CRTCA protocol since the number of active nodes is much less in CRTCA protocol than in LEACH and DSC protocols. Moreover, if there is no data packet to be sent in the CRTCA protocol, a sensor node sends a special packet to the active node at its upper level or BS to notify that it is still alive. Since the size of special packet is much smaller than that of the data packet, a special packet consumes less energy and has lower communication overhead.

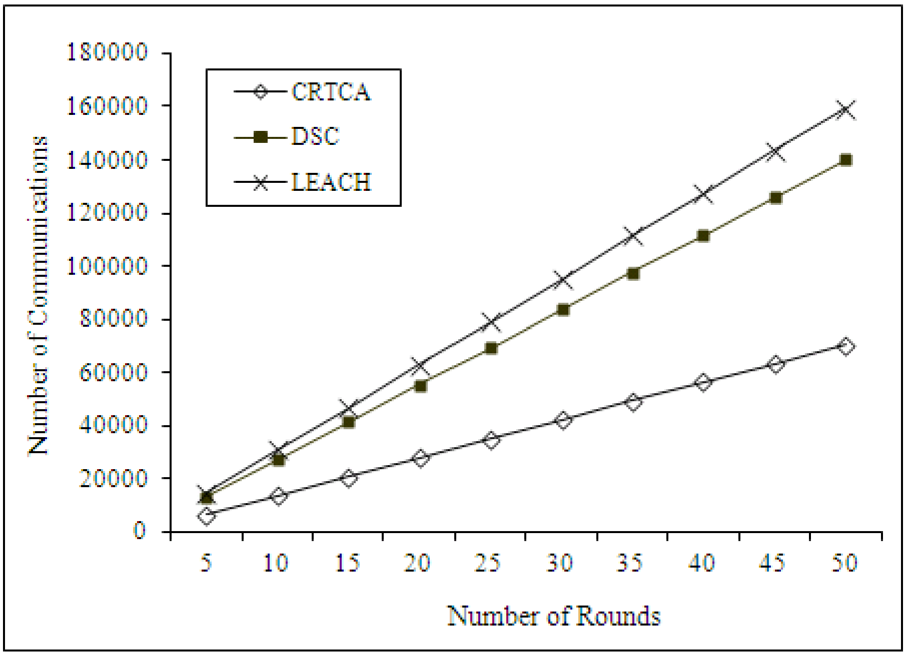

Figure 5.29: Number of communications of CRTCA protocol as compared to DSC and LEACH protocols

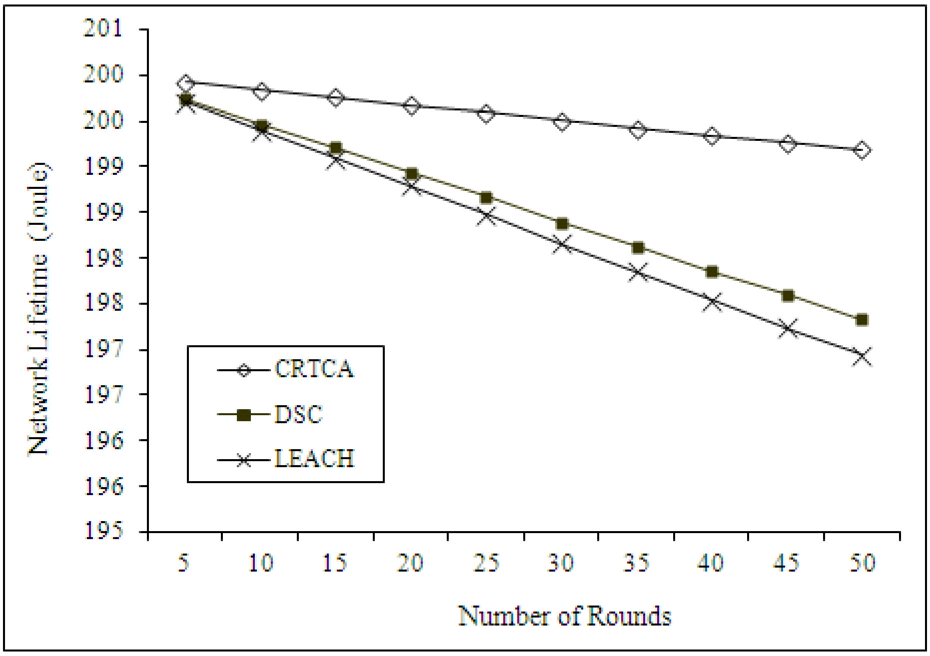

Figure 5.30: Network lifetime (or remaining network energy) of CRTCA protocol as compared to DSC and LEACH protocols

Figure 5.30 illustrates the network lifetime in terms of the remaining energy of the network over a number of rounds. We find that the energy dissipations in the LEACH and DSC protocols over a number round is much more than that in the CRTCA protocol. Hence, the LEACH and DSC protocols have shorter network life time than that in the CRTCA protocol. Student's t-test also validates this observation.

Finally, Table 5.2 presents the comparison among the CRTCA, LEACH, DSC, and some other cluster protocols on some important features.

Table 5.2: Comparison of CRTCA with existing routing protocols

Features	LEACH	TEEN	APTEEN	DSC	LESCS	CRTCA
Singlehop (directly from CH to BS)	√	X	X	√	X	X
Multi-hop(CH to BS through other nodes)	X	√	√	X	√	√
Cluster is formed by BS	X	√	√	√	√	√
Redundant data sensing	√	√	√	√	√	X
Provides fault tolerance	X	X	X	X	X	√

5.8 Summary

In this chapter, we introduced a routing protocol for Wireless Sensor Network (WSN) that combines clustering, routing, and topology control approaches. We refer to this protocol as Cluster-based Routing and Topology Control Approach (CRTCA). This protocol uses an efficient zone-based topology, and routing protocol. Moreover, a new set of graphs are used as underlying topologies for the protocol referred to as the Mini Gabriel graph (MG). Experimental results demonstrated that the CRTCA based on the proposed new set of MG graphs (CRTCA-MG) outperforms the same framework based on an existing Gabriel graph (CRTCA-GG) in terms of the network energy consumptions, transmission energy consumptions, end-to-end data transmission delay, routing delay, and number of hops in the established path. In addition, the CRTCA demonstrated the best performance in most cases over CRTCA-MG, and CRTCA-GG except the statistical analysis using student's t-test at 95% confidence level reveals that both CRCTA and CRTCA-MG have the same performance in terms of end-to-end data transmission delay for smaller value of s.

We showed through mathematical analysis, and simulation that CRTCA protocol outperforms the existing LEACH [38], and DSC [8] protocols in terms of network energy consumptions, number of data transmissions, and network lifetime. Moreover, CRTCA protocol can detect the failure of sensor nodes, data links, and packet loss and thus, provides more reliability than that in the LEACH, and DSC protocols. Experimental results also demonstrated that the CRTCA protocol balances the energy consumptions of individual sensor nodes. In the next chapter, we will present a priority-based dynamic data scheduling approach for a large scale WSN with its performance evaluation.

Chapter 6: Packet Scheduling

Among many network design issues, such as routing protocols, and data aggregation that reduce sensor's energy consumptions and data transmission delay, packet scheduling (interchangeably used as task scheduling in this chapter) is very important since it determines the transmission order of a number of data packets based on different criteria such as transmission deadline and data priority. For instance, real-time data should have the higher priority to be transmitted to the base station (BS) than non-real-time data. Thus, we introduce a Dynamic Multilevel Priority (DMP) [54] packet scheduling scheme that provides the highest priority to real-time data packets but also ensures fairness by allowing the lowest priority non-real-time data to be transmitted if they are preempted by the higher priority data for a certain period of time.

This chapter is organized as follows. Section 6.1 presents the motivation to design and implement DMP scheduling scheme, and its basic working principle. Section 6.2 defines some important terminologies that are used in designing the DMP scheduling scheme. Sections 6.3 and 6.4 present the detailed operation of DMP scheduling scheme, and its pseudo-code, respectively. Section 6.5 presents the theoretical analysis of the performance of the proposed DMP scheduling scheme in terms of end-to-end data transmission delay, and average data waiting time. Experimental results are presented in Section 6.6. Finally, summary and future work based on our experimental results are presented in Section 6.7.

6.1 Motivation

Though extensive research for scheduling the sleep/wake times of sensor nodes [5, 13, 15, 16, 33, 71, 72, 58, 89, 94, 92, 97, 109, 108, 112, 116, 128, 120] have been conducted only a few studies exist in the literature on the packet scheduling of sensor nodes [27, 88, 102, 119] that schedule the processing of data events available at a sensor node and also reduces energy consumptions. Indeed, most existing Wireless Sensor Network (WSN) applications use First Come First Serve (FCFS) [96] schedulers that process data packets in the order of their arrival time and, thus, require a lot of time to be delivered to a relevant base station (BS). However, to be meaningful, sensed data have to reach the BS within a specific time period or before the expiration of a deadline. Additionally, real-time emergency data should be delivered to BS with the shortest possible end-to-end delay. Hence, intermediate nodes require changing the delivery order of data packets in their ready queue based on their importance (e.g., real or non-real time) and delivery deadline. In non-preemptive packet scheduling schemes, real-time data packets have to wait for other already transmitting non-real-time data packets to be completed. On the other hand, in preemptive priority scheduling, lower-priority data packets can be placed into starvation for continuous arrival of higher-priority packets. In the multilevel queue scheduling algorithm [63], each node at the lowest level has a single ready queue considering that it has only local data packets to process. However, local data packets can also be real-time or non-real time and should be thus processed according to their priorities. Otherwise, emergency real-time packets may experience long queuing delays till they could be processed. Furthermore, most existing packet scheduling algorithms of WSN are neither dynamic nor suitable for large scale applications since these schedulers are predetermined and static, and cannot be changed in response to a change in the application requirements or environments. For example, for real-time applications, a real-time priority scheduler is statically used and cannot be changed during the operation of WSN applications.

Hence, we introduce a Dynamic Multilevel Priority (DMP) [54] packet scheduling scheme for WSNs, where each node maintains three levels into its queue for three different types of tasks or data packets. This is because we classify tasks or data packets as (i) real-time (highest or priority 1), (ii) non-real-time remote packets, i.e., packets that arrive from the sensors nodes at lower levels (priority 2), and (iii) non-real-time local packets, i.e., the packets that are sensed at the current sensor node (lowest or priority 3). Non-real-time data packets are classified based on the location of sensor nodes to balance the end-to-end delay of data packets that are generated at different locations. Real-time tasks can preempt tasks at other queues. If there is no task available at the real-time highest priority queue, then tasks at the second highest priority queue are processed. If the second highest priority tasks are processed at a node for α consecutive timeslots, the lowest priority tasks can then preempt the second highest priority tasks. Thus, the proposed scheduling scheme achieves fairness via an appropriate setting of α. The proposed DMP packet scheduling scheme is more suitable for heterogeneous WSN applications where both real-time and non-real time data are transmitted. For instance, it can be used in a smart home to monitor temperature and humidity (non-real time) and health condition of elderly people (real-time). Whenever a sensor sends data packets to BS through intermediate nodes, the packet type information is inserted in the packet header.

6.2 Preliminaries

In this section, we present and define some terminologies that are used in designing the Dynamic Multilevel Priority (DMP) packet scheduling scheme.

Routing Protocol

For the sake of energy efficiency and balance in energy consumptions among sensor nodes, we envision using a zone-based routing protocol [50, 47] that is presented in Chapter 5. In our proposed zone-based routing protocol, each zone is identified by a zone head (ZH) or cluster head (CH) and nodes follow a hierarchical structure, based on the number of hops they are distant from the base station (BS). For instance, nodes in zones that are one hop and two hops away from the BS are considered to be at level 1 and level 2, respectively. Each zone is also divided into a number of small squares in such a way that if a sensor node exists in square S_1, it covers all neighboring squares. Hence, if no sensor node exists in the neighboring squares, the area is still covered that reduces the probability of having any sensing hole in the network [50].

TDMA Scheme

Tasks at sensor nodes include data sensing, transmitting, receiving, and aggregation. Task processing and scheduling at each nodal level is performed using a TDMA scheme with variable-length timeslots. Data are transmitted from the lowest level nodes to BS through the nodes of intermediate levels. Thus, nodes at the intermediate/upper levels have more tasks, and processing requirements compared to lower-level nodes. Considering this observation, the length of timeslots at the upper-level nodes is set to a higher value compared with the timeslot length of lower-level nodes. On the other hand, real-time and time-critical emergency applications should stop intermediate nodes from aggregating data since they should be delivered to end users with a minimum possible delay. Hence, for real-time data, the duration of timeslots at different levels is almost equal and short.

Queue Size

It is expected that when a node x senses and receives data from the lower-level nodes, it is able to process and forward most data within its allocated timeslot. Hence, the probability that the ready queue at a node becomes full and drops packets is low. However, if any data packet remains in the ready queue of node x during its allocated timeslot, that data packet will be processed in the next allocated timeslot. If the scheduling scheme uses a multichannel MAC protocol, nodes can send multiple packets to other nodes.

Fairness

This metric ensures that data of different priorities get carried out with a minimum waiting time at the ready queue based on the priority of data. For instance, if any lower-priority data waits for a long period of time for the continuous arrival of higher-priority data, fairness defines a constraint that allows the lower-priority data to get processed after a certain waiting time.

Priority

As discussed earlier, real-time and emergency data should have the highest priority. The priority of data can be also assigned based on the size of the data. The shortest data packets have the highest priority. However, if it is observed that the size of data packets is always smaller than the data packets at the lower priority queue, they are preempted to allow low-priority data packets to be processed after a certain waiting period. Nevertheless, these tasks can be preempted by real-time emergency tasks.

6.3 Dynamic Multilevel Priority (DMP) Packet Scheduling

Scheduling data packets among several queues of a sensor node is illustrated in Figure 6.1. Data packets that are sensed at a node are scheduled among a number of levels in the ready queue. Then, a number of data packets in each level of the ready queue is scheduled. For instance, Figure 6.1 demonstrates that the data packet, Data1 is scheduled to be placed in the first level, Queue1. Then, Data1 and Data3 of Queue1 are scheduled to be transmitted based of different criteria. The general working principle of the proposed DMP scheduling scheme is illustrated in Figure 6.2.

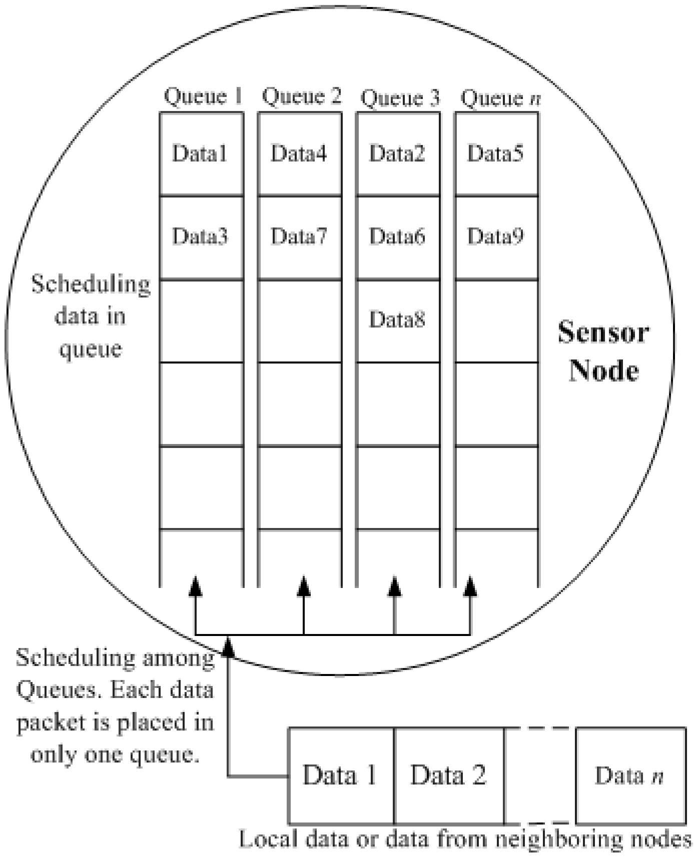

Figure 6.1: Scheduling data among multiple queues

The proposed scheduling scheme assumes that nodes are virtually organized following a hierarchical structure. Nodes that are at the same hop distance from the base station (BS) are considered to be located at the same level. Data packets of nodes at different levels are processed using the Time-Division Multiplexing Access (TDMA) scheme. For instance, nodes that are located at the lowest level and the second lowest level can be allocated timeslots 1 and 2, respectively. We consider three-level of queues, that is, the maximum number of levels in the ready queue of a node is three: priority 1 (pr_1), priority 2 (pr_2), and priority 3 (pr_3) queues. Real-time data packets go to pr_1, the highest priority queue, and are processed using FCFS.

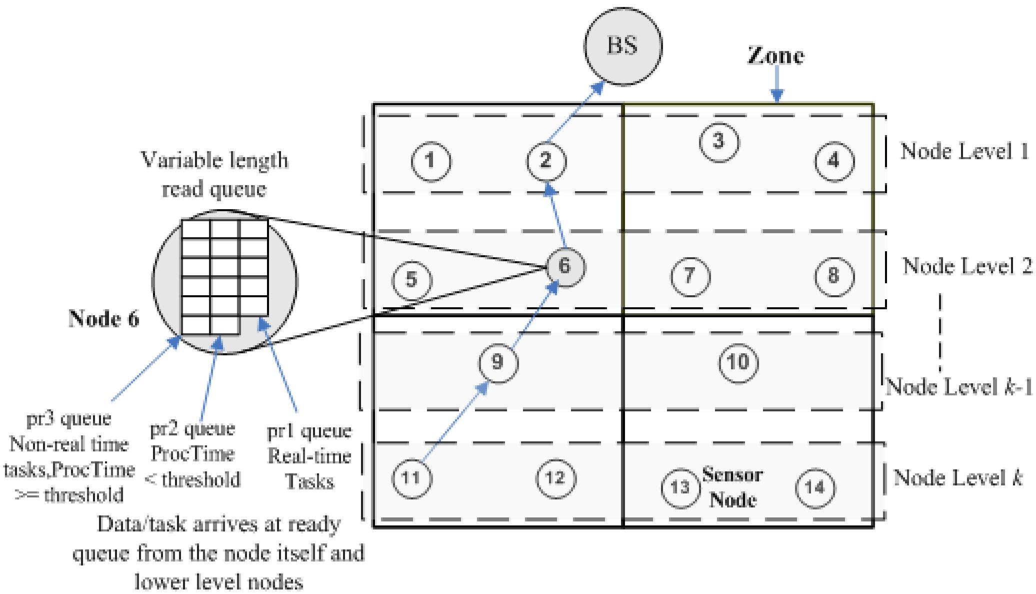

Figure 6.2: Proposed dynamic multi-level priority (DMP) packet scheduling scheme

Non-real-time data packets that arrive from sensor nodes at lower levels go to pr_2, the second highest priority queue. Finally, non-real time data packets that are sensed at a local node go to pr_3, the lowest priority queue. The possible reasons for choosing maximum three queues are to process (i) real-time pr_1 tasks with the highest priority to achieve the overall goal of WSNs, (ii) non real-time pr_2 tasks to achieve the minimum average task waiting time and also to balance the end-to-end delay by giving higher priority to remote data packets, (iii) non-real-time pr_3 tasks with lower priority to achieve fairness by preempting pr_2 tasks if pr_3 tasks wait a number of consecutive timeslots.

In the proposed scheme, queue sizes differ based on the application requirements. Since preemptive priority scheduling incurs overhead due to the context storage and switching in resource constraint sensor networks, the size of ready queue for non-preemptive priority scheduler is expected to be smaller than that of the preemptive priority scheduler. The idea behind this is that the highest priority real-time emergency tasks are processed with a minimum possible delay. They are placed in non-preemptive priority tasks queue (pr_1 queue) and can preempt the currently running tasks. Thus, they are expected not to reserve a queue location for a longer period of time. Moreover, the number of preemptions will be low since the number of real-time data packets is in general small in number. On the other hand, non-real-time packets that arrive from the sensor nodes at lower level are placed in the preemptive priority queue (pr_2 queue). The processing of these data packets can be preempted by the highest priority real-time tasks and also after a certain time period if tasks at the lower priority pr_3 queue do not get processed due to the continuous arrival of higher priority data packets. Real-time packets are usually processed in FCFS fashion. Each packet has an ID, which consists of two parts, namely level ID and node ID. When two equal priority packets arrive at the ready queue at the same time, the data packet which is generated at the lower level will have higher priority. This phenomenon reduces the end-to-end delay of the lower level tasks to reach the BS. For two tasks of the same level, the smaller task (i.e., in terms of data size) will have higher priority.

Timeslots at each level are not fixed. They are rather calculated based on the data sensing period, data transmission rate, and CPU speed. They are increased as the levels progress through BS. However, if there is any real-time or emergency response data at a particular level, the time required to transmit that data will be short and will not increase at the upper levels since there is no data aggregation. The remaining time of a timeslot of nodes at a particular level will be used to process data packets at other queues. Since the probability of having real-time emergency data is low, it is expected that this scenario would not degrade the system performance. Instead, it may improve the perceived Quality of Service (QoS) by delivering real-time data fast. Moreover, if any node x at a particular level completes its task before the expiration of its allocated timeslot, node x goes to sleep by turning its radio off for the sake of energy efficiency.

6.4 Pseudo-code

In our proposed DMP task scheduling scheme, nodes at the lowest level, l_k, sense, process and transmit data during their allocated timeslots, whereas nodes at level l_{k-1} and upper levels receive data in addition to sensing, processing and transmitting data. Now, we present the pseudo-code of our proposed DMP task scheduling scheme.

---

while $task_{k,i}$ is received by $node_i$ at level k, i.e., l_k **do**

  if $Type(task_{k,i}) = real - time$ **then**

    put $task_{k,i}$ into pr_1 queue

  else if $node_i$ is not at lowest levels **then**

    if $task_{k,i}$ is not local **then**

      put $task_{k,i}$ into pr_2 queue

    else

      put $task_{k,i}$ into pr_3 queue

    end if

  else

    put $task_{k,i}$ into pr_2 queue

end if

Assume, the duration of a timeslot at $l_k \leftarrow t(k)$

Data sensing time of $node_i$ at $l_k \leftarrow senseTime_k(t)$

Remaining time after data sensing, $t_1(k) = t(k) - senseTime_k(t)$

Let total real-time tasks for $node_i$ at $l_k \leftarrow n_k(pr_1)$

Let $procTimepr_1(k) \leftarrow \sum_{j=1}^{n_k(pr1)} procTime(j)$

if $procTimepr1(k) < t_1(k)$ **then**

  All pr_1 tasks of $node_i$ at l_k are processed as FCFS

  Remaining time $t_2(k) \leftarrow t_1(k) - procTimepr_1(k)$

  Let, total pr_2 tasks for $node_i$ at $l_k \leftarrow n_k(pr_2)$

  Let $procTimepr_2(k) \leftarrow \sum_{j=1}^{n_k(pr2)} procTime(j)$

  if $procTimepr_2(k) < t_2(k)$ **then**

    All pr_2 tasks are processed as FCFS

    pr_3 tasks are processed as FCFS for the remaining time, $t_3(k) \leftarrow t_2(k) - procTimepr_2(k)$

  else

    pr_2 tasks are processed for $t_2(k)$ time

    no pr_3 tasks are processed

  end if

else

  only pr_1 tasks are processed for $t_1(k)$ time

  no pr_2 and pr_3 tasks are processed

end if

if pr_1 queue empty pr_2 tasks are processed α consecutive timeslots since $t(k) \leq procTimepr_2(k)$ **then**

  pr_2 tasks are preempted at $\alpha + 1, \ldots, \alpha + j$ timeslots by pr_3 tasks

if pr_1 task arrives during any of $\alpha+1, \alpha+2, \ldots, \alpha+j$ timeslots **then**

pr_3 tasks are preempted and pr_1 tasks are processed

context are transferred again for processing pr_3 tasks

end if

end if

end while

We consider only two levels in the ready queue of sensor nodes that are located at the lowest level since these nodes do not receive packets from any lower level nodes. Other nodes have three levels in the ready queue and place non-real time local tasks into pr_3 queue. We also consider that each node requires time to sense data packets and also process local and/or remote data packets. For instance, $t_1(k)$ in the pseudo-code represents the real-time data sensing time at a $node_i$. If the processing time of real-time data at $node_i$ is less than $t_1(k)$ then $node_i$ will have time remaining to process non-real-time pr_2 data packets. Similarly, if $node_i$ still has some remaining time, it can process non-real-time pr_3 data packets. The pseudo-code also shows that if the pr_1 queue is empty and pr_2 packets are processed α consecutive timeslots, the processing of pr_2 data packets will be preempted for j timeslots.

6.5 Performance Analysis

In this section, we analyze the performance of the proposed DMP task scheduling scheme in terms of end-to-end delay, and total waiting time of different types of traffic at the ready queues of active nodes.

6.5.1 End-to-End Delay

In the following, we formulate the average end-to-end delay of transmitting different priority data packets to the base station (BS). Again, we interchangeably use task and data to represent the data packets that are sensed at a sensor node.

Real-time Priority 1 Queue Data

Let us assume that a node x, residing at level l_k is sensing a real-time, emergency event, e.g., fire detection. This node transmits the emergency priority 1 data to BS through l_{k-1} intermediate levels. We consider the following scenario whereby every time a real-time data packet reaches a neighboring active node, y at an upper level, a non-real time lower priority data is being processed at that node. Hence, data delivery at y is preempted to send real-time data.

Transmission time or delay that is required to place a real-time data from a node into the medium is equal to $\frac{data_{pr1}}{s_t}$. The propagation time, or delay to transmit data from the source to destination can be formulated as $\frac{d}{s_p}$. Considering the above mentioned scenario the end-to-end delay for sending a real-time data satisfies the following inequality.

$$delay_{pr1} \geq l_k \times (\frac{data_{pr1}}{s_t} + pr1_{proc}(t)) + \frac{d}{s_p} + (l_k \times t_{overhead}) \tag{6.1}$$

where $data_{pr1}$ denotes the real-time data size, s_t denotes the data transmission speed, d is the distance from the source node to BS, where $d = \sum_{i=1}^{l_k} d_i$, s_p denotes the propagation speed over the wireless medium, $pr1_{proc}(t)$ is the processing time of real-time tasks at each node, and $t_{overhead}$ is an overhead in terms of context switching and queuing time (including time for preemption). However, a real-time task t_1 has to wait if there is a number, n_{pr1}, of a real-time task ahead of t_1 at the pr_1 queue. We assume that all real-time data have the same size.

Therefore, the end-to-end delay for a real-time task t_1 considering that t_1 has n_{pr1} number of real-time tasks ahead of it,

$$delay_{t_1} \geq \sum_{i=1}^{n_{pr1}} (delaypr_1)_i \tag{6.2}$$

Non-real time Priority 2 Queue Data

Tasks at pr_2 queue can be preempted by real-time ones. Taking the scenario of Figure 6.2 as an example, we first consider the scenario when a real-time task is sensed at node 11 and is forwarded to BS through relay nodes 9, 6, and 2. It should be observed that tasks are available at the pr_2 queue at nodes 9, 6 and 2. Since one real-time task is available at the pr_1 queue of nodes 9, 6, and 2, real-time tasks will be processed and transmitted first during the timeslot of nodes 9, 6, and 2. The pr_2 tasks are processed in the remaining time of the timeslots. The transmission time or delay to place pr_2 data from a node into the medium can be therefore computed as $\frac{data_{pr2}}{s_t}$. Thus, the total end-to-end delay for a pr_2 task that can be processed in the same timeslot exceeds

$$l_k \times (\frac{data_{pr1}}{s_t} + \frac{data_{pr2}}{s_t} + pr1_{proc}(t) + pr2_{proc}(t)) + \frac{d}{s_p} + (l_k \times t_{overhead}) \tag{6.3}$$

Non-real time Priority 3 Queue Data

In the best case, when no task is available at the pr_1 and pr_2 queues, the end-to-end delay of the pr_3 tasks will be almost equal to that of the pr_1 queue tasks (Equation 6.1) although it can differ slightly based on the size of the pr_3 queue task. We assume that the pr_3 queue tasks are processed by preempting pr_2 queue tasks if for α consecutive timeslots there is no task at the pr_1 queue but there are tasks available at the pr_2

queue. Let t_k denote the length of a timeslot of nodes at level l_k. The transmission time or delay to place pr_3 data from a node into the wireless medium is equal to $\frac{data_{pr3}}{s_t}$. However, during the processing of the pr_3 queue tasks, these tasks can be preempted by real-time tasks. They are processed again after the completion of real-time tasks. Thus, the end-to-end delay for processing pr_3 tasks will be exceeding

$$\alpha \times t(k) + l_k \times (\frac{data_{pr3}}{s_t} + pr3_{proc}(t)) + \frac{d}{s_p}) + (l_k \times t_{overhead}) \tag{6.4}$$

6.5.2 Average Waiting Time

In the following, we formulate the average waiting time of tasks at different workloads. Let us assume that pr_{ij_i} represents the processing time of the j-th pr_i task at a node x, where, $1 \leq i \leq 3$ and $1 \leq j_i \leq n_i$.

Thus, total processing time, $pr_i(t) = \sum_{j_i=1}^{n_i} pr_{ij_i}(t)$. Let us denote the total number of levels as k, and the length of a timeslot at the level l_j as $t(j)$.

For real-time tasks, $i = 1$ (i.e., pr_1). Assuming that real-time and emergency tasks rarely occur and require a very short time to get processed, $pr_1(t) < t(k)$. Hence, all tasks,$1 \leq j_1 \leq n_1$, in the pr_1 queue complete processing and tasks in the pr_2 and pr_3 queues are processed for the remaining, $t_2(k) = t(k) - pr_1(t)$, period of time.

Since pr_1 tasks are processed as FCFS, the average waiting time for real-time, pr_1 tasks at node x is

$$AvgWaitingTimePr_1(t) = \frac{\sum_{j_1=1}^{n_1-1} \sum_{m=1}^{j_1} pr_{1,m}(t)}{n_1} \tag{6.5}$$

where the first pr_1 task has no waiting time and waiting time for the j-th pr_1 task is equal to $\sum_{m=1}^{j} pr_{1,m}(t)$. Now, let pr_2 tasks be sorted according to the ascending order of the processing time, $pr_{2j_2}(t)$, of pr_2 tasks at the ready queue so that we have $pr_{21}(t) \leq pr_{22}(t) \leq \ldots pr_{2n}(t)$. If pr_2 tasks are not preempted by pr_1 tasks and can be completed within the $t_2(k)$ time (i.e., within the same timeslot for the processing pr_1 tasks), the average waiting time for pr_2 tasks can be expressed as follows:

$$AvgWaitingTimePr_2(t) = \frac{\sum_{j_2=1}^{n_2-1}\sum_{m=1}^{j_2} pr_{2,m}(t)}{n_2} \tag{6.6}$$

If pr_2 tasks at i-th node at the level j require more than one timeslot to complete their processing,

$frameTime = \sum_{j=1}^{k} t(j)$, $node_{ij}$ waits $\tau = (\sum_{j=1}^{k} t(j)) - t(j)$.

If pr_2 queue tasks at $node_{ij}$ requires $\beta \geq 1$ timeslots to complete their processing and no pr_1 task preempts pr_2 tasks, then the average waiting time of pr_2 tasks is

$$\begin{aligned}AvgWaitingTimePr_2(t) \approx \frac{\sum_{j_{21}=1}^{n_{21}}\sum_{m=1}^{j_{21}} pr_{2,m}(t)}{n_{21}} + \frac{\sum_{j_{22}=n_{21}+1}^{n_{22}}\sum_{m=1}^{j_{22}} pr_{2,m}(t)}{n_{22}} + \dots \\ + \frac{\sum_{j_{2,\beta-1}=n_{(2,\beta-2)}+1}^{n_{2,\beta-1}}\sum_{m=1}^{j_{2,\beta-1}} pr_{2,m}(t)}{n_{2,\beta}} + (\beta \times \tau)\end{aligned} \tag{6.7}$$

Where, $n_{21} + n_{22} + \ldots + n_{2,\beta} = n_2$ and $\beta \times \tau$ is the total waiting time of pr_2 tasks to work for β timeslots. Again, τ is the waiting time of pr_2 tasks of a node at a specific level to complete the remaining work in the next timeslot. Moreover, there is no other waiting time, even if a new pr_2 task arrives while processing other pr_2 tasks, since pr_2 tasks are non-preemptive (until a new pr_1 task preempts pr_2 tasks). In general, the average waiting time of a node for processing pr_2 tasks considering preemption by pr_1 tasks is

$$\begin{aligned}AvgWaitingTimePr_2(t) \geq \frac{\sum_{j_{21}=1}^{n_{21}}\sum_{m=1}^{j_{21}} pr_{2,m}(t)}{n_{21}} + \frac{\sum_{j_{22}=n_{21}+1}^{n_{22}}\sum_{m=1}^{j_{22}} pr_{2,m}(t)}{n_{22}} + \dots \\ + \frac{\sum_{j_{2,\beta-1}=n_{(2,\beta-2)}+1}^{n_{2,\beta-1}}\sum_{m=1}^{j_{2,\beta-1}} pr_{2,m}(t)}{n_{2,\beta}} + (\beta \times \tau) + \sum_{m=1}^{\beta}\gamma_m\end{aligned} \tag{6.8}$$

where $1 \leq m \leq \beta$ and γ_m are the extra waiting time of pr_2 tasks for being preempted by pr_1 tasks in each of the m-th timeslots. If we consider that, at each of the timeslots, pr_2 tasks will be processed after processing pr_1 tasks for the $pr_1(t) = \sum_{j_i=1}^{n_1} pr_{1j_1}(t)$ period, then the processing time of pr_1 tasks will be considered as the waiting time of pr_2 tasks at each of the timeslot.

$$AvgWaitingTimePr_2(t) \geq \frac{\sum_{j_{21}=1}^{n_{21}} \sum_{m=1}^{j_{21}} pr_{2,m}(t)}{n_{21}} + \frac{\sum_{j_{22}=n_{21}+1}^{n_{22}} \sum_{m=1}^{j_{22}} pr_{2,m}(t)}{n_{22}} + \dots + \frac{\sum_{j_{2,\beta-1}=n_{(2,\beta-2)}+1}^{n_{2,\beta-1}} \sum_{m=1}^{j_{2,\beta-1}} pr_{2,m}(t)}{n_{2,\beta}} + (\beta \times \tau) + \sum_{m=1}^{\beta} \gamma_m + (\beta \times pr_1(t)) \tag{6.9}$$

Now, consider that pr_2 tasks are preempted by pr_3 tasks if no pr_1 tasks exist at the ready queue and pr_2 tasks are processed for $\alpha \geq 2$ consecutive timeslots. If $\beta \leq \alpha$, then the average waiting time of pr_2 tasks will be the same as Equation 6.9. If $\beta > \alpha$ and the task processing controller are switched back to pr_2 tasks from pr_3 tasks after δ timeslots, the average waiting time of pr_2 tasks,

$$AvgWaitingTimePr_2(t) \geq \frac{\sum_{j_{21}=1}^{n_{21}} \sum_{m=1}^{j_{21}} pr_{2,m}(t)}{n_{21}} + \frac{\sum_{j_{22}=n_{21}+1}^{n_{22}} \sum_{m=1}^{j_{22}} pr_{2,m}(t)}{n_{22}} + \dots + \frac{\sum_{j_{2,\beta-1}=n_{(2,\beta-2)}+1}^{n_{2,\beta-1}} \sum_{m=1}^{j_{2,\beta-1}} pr_{2,m}(t)}{n_{2,\beta}} + (\beta \times \tau) + \sum_{m=1}^{\beta} \gamma_m + (\beta \times pr_1(t) + (\delta \times \sum_{j=1}^{k} t(j)) \tag{6.10}$$

Equation 6.10 presents the waiting time of pr_2 tasks of the nodes at upper levels and have pr_1, pr_2 and pr_3 queues at each node. But the lowest-level nodes only have the pr_1 and pr_2 queues; therefore, pr_2 tasks are not preempted by pr_3 tasks at the lowest level. Similarly, we can formulate the average waiting time of pr_3 tasks of nodes at the upper level as follows.

If pr_3 tasks at a node get processed after α timeslots or frames due to the processing of pr_1 and pr_2 tasks, then pr_3 tasks are processed after waiting for $\alpha \times \sum_{j=1}^{k} t(j)$ time period. Then we assume that the pr_3 tasks require ψ timeslots to complete their tasks and, during these timeslots, the pr_3 tasks are preempted by pr_1 tasks for $\sum_{m=1}^{\psi} \gamma_m$ period.

Thus, the average waiting time of pr_3 tasks at a node, $AvgWaitingTimePr3(t)$, exceeds

$$AvgWaitingTimePr_3(t) \geq \frac{\sum_{j_{31}=1}^{n_{31}} \sum_{m=1}^{j_{31}} pr_{3,m}(t)}{n_{31}} + \frac{\sum_{j_{32}=n_{31}+1}^{n_{32}} \sum_{m=1}^{j_{32}} pr_{3,m}(t)}{n_{32}} + \ldots$$
$$+ \frac{\sum_{j_{3,\psi-1}=n_{(3,\psi-2)}+1}^{n_{3,\psi-1}} \sum_{m=1}^{j_{3,\psi-1}} pr_{3,m}(t)}{n_{3,\psi}} + (\psi \times \tau) + \sum_{m=1}^{\psi} \gamma_m + (\alpha \times \sum_{j=1}^{k} t(j)) \tag{6.11}$$

where $n_{31} + n_{32} + \ldots + n_{3,\psi} = n_3$ (i.e., the total number of pr_3 tasks at a node). Thus, using Equations 6.6 - 6.11 we formulate the average waiting time of pr_1, pr_2, and pr_3 tasks at the ready queue of a node x at a particular level.

6.6 Performance Evaluation

The simulation model is implemented using the C programming language. It is used to evaluate the performance of the proposed DMP packet scheduling scheme, comparing it against the FCFS, and Multilevel Queue scheduling schemes. The comparison is made in terms of average packet waiting time, and end-to-end data transmission delay. We use randomly connected Unit Disk Graphs (UDGs) on a surface of 100 meter $\times$ 100 meter as a basis of our simulations. The number of simulated zones varies from 4 to 12 zones. Nodes are distributed uniformly over the zones. The ready queue of each node can hold a maximum of 50 tasks. Each task has a Type ID that identifies its type. For instance, type 0 is considered to be a real-time task. Data packets are placed into the ready queue based on the processing time of the task. Moreover, each packet has a hop count number that is assigned randomly, and the packet with the highest hop count number is placed into the highest-priority queue. We run the simulation both for a specific number of zones, and levels in the network until data from a node in each zone or level reach BS. Simulation results are presented for both real-time data and all types of data traffic. Table 6.1 presents simulation parameters, and their respective values.

Table 6.1: Simulation parameters, and their respective values

Parameter	**Value**
Network Size	100m X 100m
Number of Nodes	Maximum 200
Number of Zones	4 - 12
Base station position	55m X 101m
Transmission Energy Consumptions	50 $nJoule/bit$
Energy Consumption in free space or air	0.01 $nJoule/bit/m^2$
Initial Node Energy	2 Joule
Transmission Speed	250$Kbps$
Propagation Speed	$198 \times 10^6 meter/sec$

Figures 6.3 and 6.4 illustrate the end-to-end data transmission delay of real-time tasks over a number of zones and levels, respectively. In both case, we observe that the proposed DMP scheduling scheme outperforms the existing FCFS, and Multi-level Queue scheduler. This is because the proposed scheduling scheme gives the highest priority to real-time tasks and also allows real-time data packets to preempt the processing of non-real time data packets. Thus, real-time data packets have lower

data transmission delays.

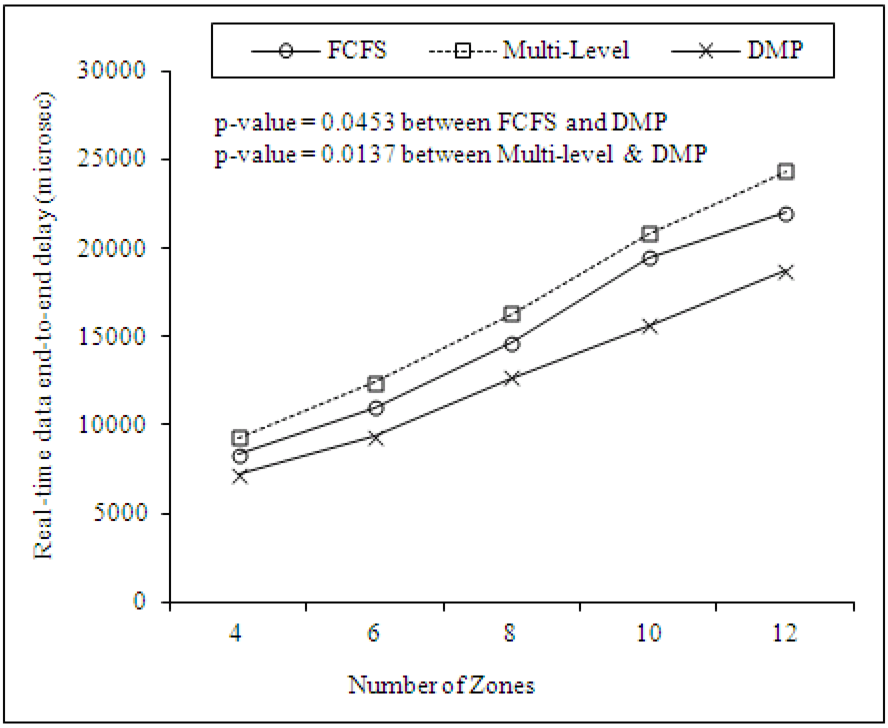

Figure 6.3: End-to-end delay of real-time data over a number of zones

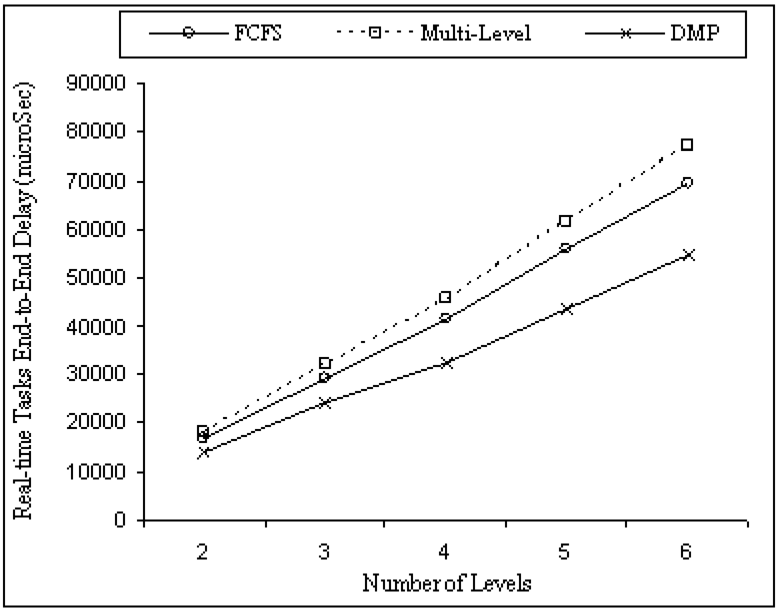

Figure 6.4: End-to-end delay of real-time data over a number of levels

We also validate these results using student's t-test at 95% confidence level. Figure 6.3 illustrates the p-values which are 0.0453 between FCFS and DMP schemes and 0.0137 between Multi-level queue and DMP schedulers.

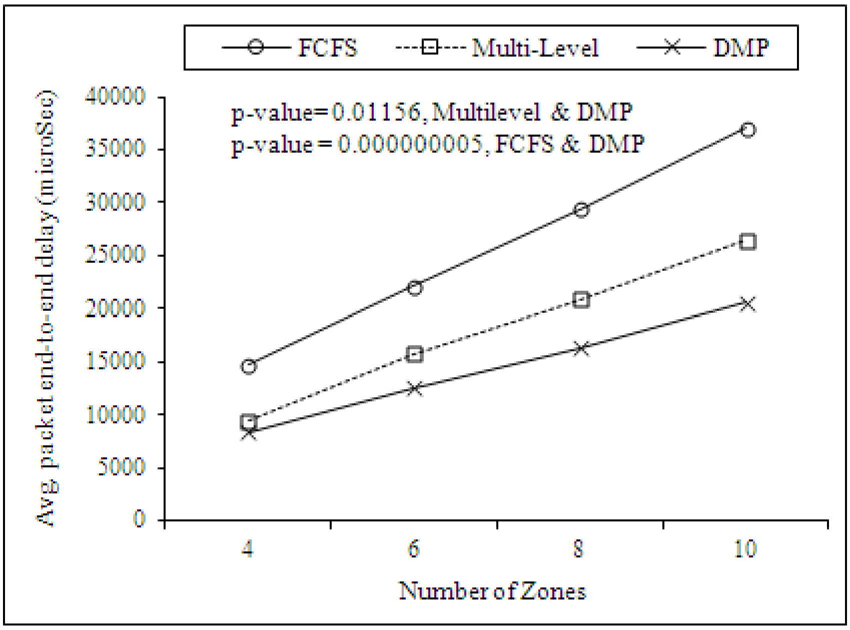

Figure 6.5: End-to-end delay of all types of data over a number of zones

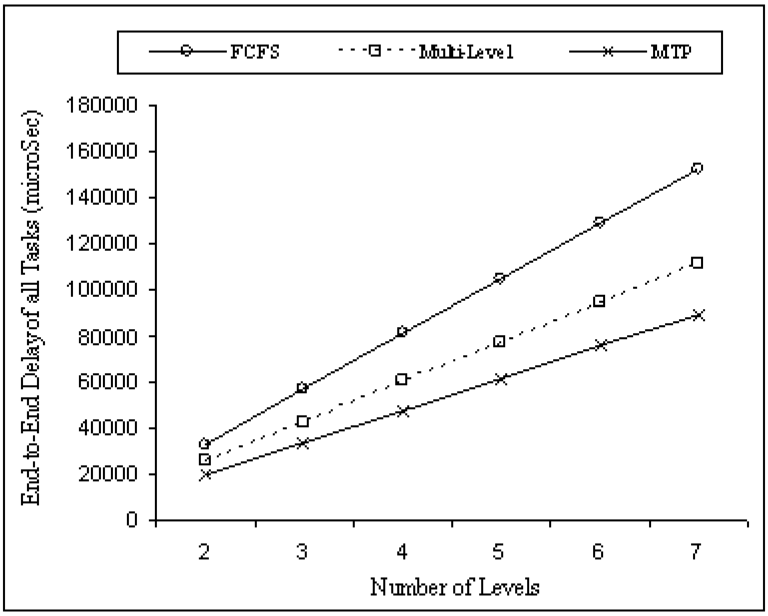

Figure 6.6: End-to-end delay of all types of data over a number of levels

Similarly, Figures 6.5 and 6.6 demonstrate the end-to-end delay of all types of data traffic over a number of zones and levels, respectively. From these results, we find that the DMP task scheduling scheme outperforms FCFS, and Multilevel Queue scheduler in terms of end-to-end data transmission delay. This is because in the proposed scheme, the tasks that arrive from the lower level nodes are given higher priority than the tasks at the current node. Thus, the average data transmission delay is shortened. Figure 6.5 shows the p-values of student's t-test, which are 0.01156 between Multi-level and DMP schedulers, 0.000000005 between FCFS and DMP schedulers. Thus, DMP outperforms both FCFS and Multi-level queue schedulers at 95% confidence interval.

Figures 6.7 - 6.10 demonstrate that the DMP task scheduler has better performance than the FCFS, and Multilevel Queue scheduler in terms of average task waiting time, both for real-time tasks, and all types of tasks. We have already explained the possible reasons for this performance differences. We also perform student's t-test at a 95% confidence level and find the p-value to be less than 0.05 in most cases. This test validates our claim about the performance of the proposed DMP scheduling scheme.

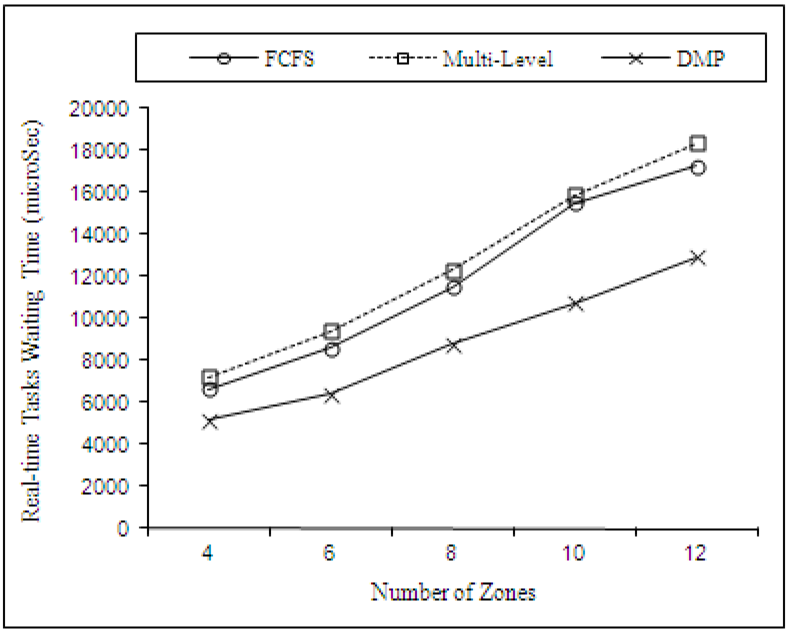

Figure 6.7: Waiting time of real-time data over a number of zones

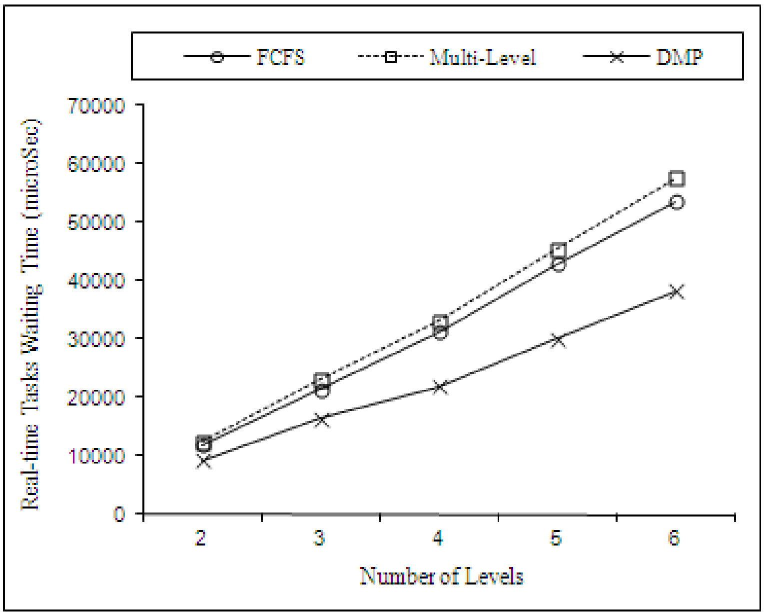

Figure 6.8: Waiting time of real-time data over a number of levels

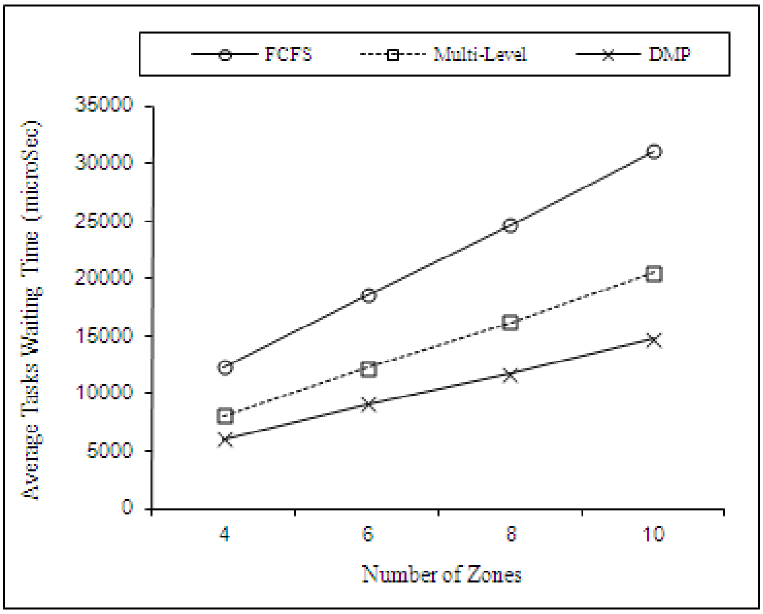

Figure 6.9: Waiting time of all types of data over a number of simulated zones

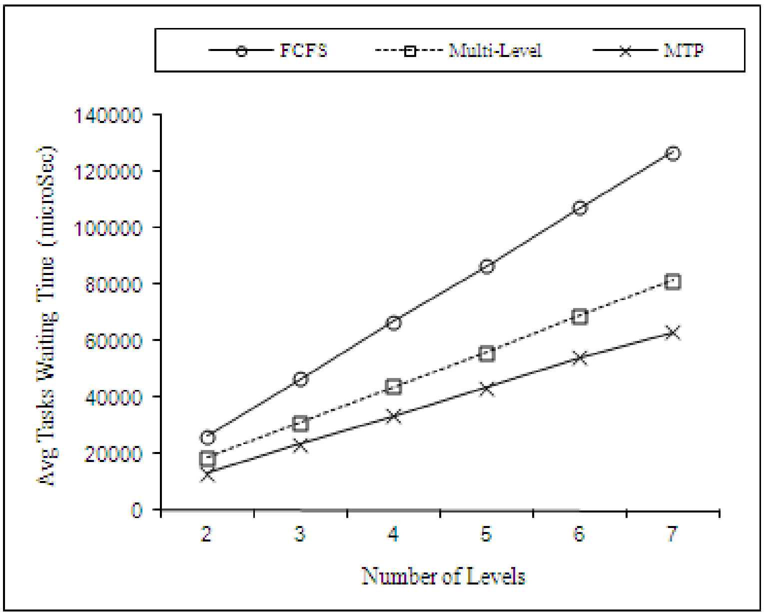

Figure 6.10: Waiting time of all types of data over a number of levels

We also measure, and compare the fairness of executing non-real-time task in terms of the total waiting time of non-real-time tasks over total waiting time of all tasks. Figure 6.11 illustrates that the fairness index of DMP scheduling scheme is higher or better than that of the other two approaches. The number of levels in the network topology increases as the number of zones multiplies, which increases the average waiting time for non-real-time tasks over real-time tasks. Thus, the fairness index slightly decreases or remains almost same as the number of zones increases.

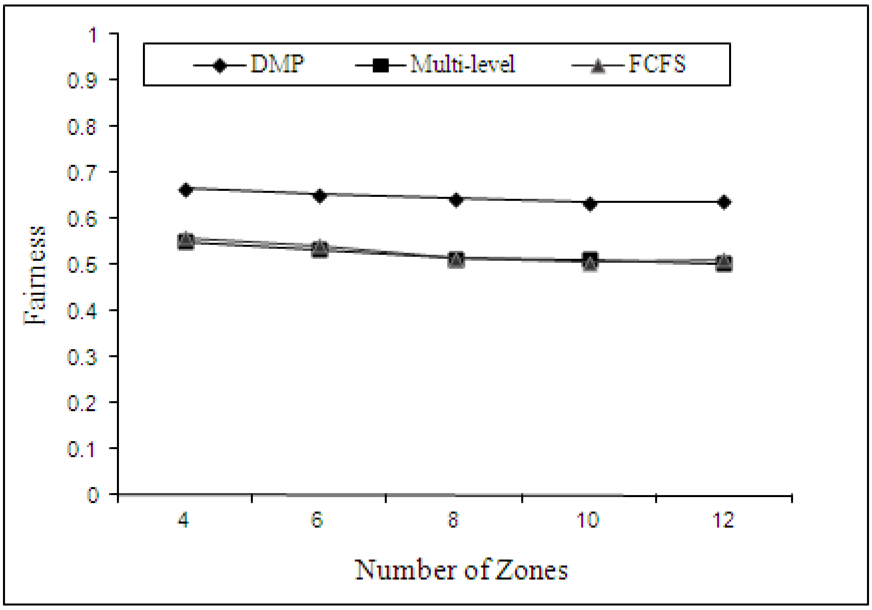

Figure 6.11: Fairness in terms of the waiting time of non-real-time data

Using the concept of three-level priority queues at each node, the proposed DMP task scheduling scheme allows different types of data packets to be processed based on their priorities. Since real-time, and emergency data should be processed with the minimum end-to-end delay, they are processed with the highest priority, and can preempt tasks with lower priorities located in the two other queues. On the other hand, in existing multilevel queue schedulers, a task with the highest hop count is given the highest priority. Hence, real-time tasks are prioritized over other task types only if their hop counts are higher than those of non-real-time tasks. Moreover, in FCFS and multilevel queue schedulers, the estimated processing time of a task is not considered when deciding the priority of a task. Thus, FCFS and Multilevel Queue schedulers exhibit longer task waiting times and end-to-end delays, in comparison to the DMP task scheduling scheme. Furthermore, the average waiting time of a task contributes largely to the experienced end-to-end data transmission delay, hence the strong correlation between the results of Figures 6.9 and 6.6.

In the DMP task scheduling approach, the source of a data packet is used to define the priority of data packets other than real-time. The priority of non-real time data packet will be more if it is sensed at remote node rather than the current sending node. Moreover, when no real-time tasks are available, pr_3 tasks can preempt pr_2 tasks if they are in starvation for a long time. This allows the processing of different types of tasks with fairness. The memory is also dynamically allocated to three queues and the size of the highest-priority queue is usually smaller than the two other queues (Figure 6.2) since pr_1 real-time tasks do not occur frequently compared to non-real-time tasks. As the memory capacity of a sensor node is limited, this also balances memory usages. Moreover, tasks are mostly non-real-time and are processed in the pr_2 and pr_3 queues. Non-real-time tasks that a node x receives from the lower level nodes are known as non-real-time remote tasks and processed with higher priority (pr_2) than the non-real-time local tasks that x senses. Thus, non-real-time remote tasks incur less average waiting time. In addition, the average waiting time will not be affected for real-time tasks that are processed using FCFS scheduling, since these real-time tasks occur infrequently with a short processing time.

Admittedly, one of the concerns regarding our proposed DMP task scheduling scheme pertains to its energy requirements. Indeed, the DMP task scheduling mechanism could be less energy efficient in comparison to the other two approaches since the DMP scheme requires a few more processing cycles to categorize and place the tasks into three different queues as well as for context saving and switching (for preemption). However, given the increased demand for WSN-based solutions that efficiently support real-time emergency applications and ensure them minimum average task waiting time and end-to-end delay, the proposed DMP task scheduling mechanism can be regarded as highly efficient.

6.7 Summary

In this chapter, we introduced a Dynamic Multilevel Priority (DMP) task scheduling scheme for WSNs. The scheme uses three-level of priority queues to schedule tasks based on their types, and priorities. It ensures minimum end-to-end data transmission delay for the highest priority real-time tasks while exhibiting acceptable fairness towards lowest-priority non-real time local tasks. Experimental results showed that the proposed DMP task scheduling scheme has better performance than the existing FCFS, and Multilevel Queue Scheduler scheme in terms of the average task waiting time, and end-to-end delay.

As enhancements to the proposed DMP scheme, we envision assigning task priority based on task deadline instead of the arrival order or the shortest task processing time. To reduce processing overhead and save bandwidth, we could also consider removing tasks with expired deadlines from the medium. Furthermore, if a real-time task holds the resources for a longer period of time, other tasks need to wait for an undefined period of time, causing the occurrence of a deadlock. This deadlock situation degrades the performance of task scheduling schemes in terms of end-to-end delay. Hence, we would deal with the circular wait, and preemptive conditions [96] to prevent deadlock from occurring. In the next chapter, we will introduce an energy efficient and dynamic data aggregation approach for a large scale WSN.

Chapter 7: Data Aggregation

In a large scale Wireless Sensor Network (WSN), sensors are deployed in a large number that sense redundant data. Transmitting these redundant data over the network results traffic congestion, increases sensors energy consumptions, and end-to-end data transmission delay. Thus, data aggregation is used to reduce the number of redundant data at intermediate sensor nodes, and end-to-end data transmission delay. However, most existing data aggregation techniques [18, 41, 42, 46, 45, 61, 117] are not designed for large scale WSNs and also do not consider the importance of real-time data while aggregating traffic of different types. Thus, we introduce a dynamic data aggregation technique for a large scale WSN that provides the highest priority to real-time data during aggregation and trades-off energy efficiency, and end-to-end data transmission delay.

This chapter is organized as follows. Section 7.1 presents the general working principle of the proposed data aggregation approach with its pseudo-code. Section 7.2 presents the theoretical analysis of the proposed data aggregation approach in terms of aggregated data size, network energy consumptions, and data propagation time. Section 7.3 presents simulation results. Section 7.4 discusses the performance of the proposed data aggregation approach, and compares with existing approaches on some important features. A summary of this chapter is presented in Section 7.5.

7.1 Data Aggregation

Once paths are established, and data packets that are sensed by or arrive at sensor nodes are scheduled they (packets) are routed through the nodes of a path from the source node to base station (BS). We assume that nodes in different zones are located at different levels of a virtual hierarchy. The level number of a node is determined using the number of hops the node is away from BS.

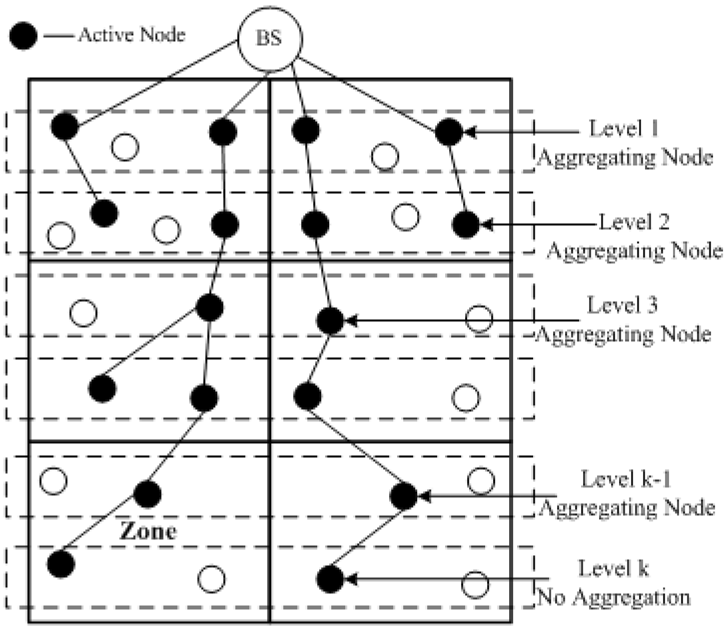

Figure 7.1: Data aggregation tree

Figure 7.1 illustrates the zone-based topology, and virtual hierarchy. Active nodes at different levels work using TDMA scheme, where the length of timeslots at different levels are variable. For instance, at timeslot 1, the active nodes at the farthest level L_k sense the subscribed events and send corresponding data to the active nodes at upper level L_{k-1}. If a node x at the level L_k does not have any subscribed event to send at the timeslot 1, the node x sends a small sized special packet to the node at the upper level with which x has an established path. At timeslot 2, active nodes at the level L_{k-1} receive data or special packet, send acknowledgement to nodes at the level L_k and also transmit data or special packet to nodes at the upper level L_{k-2}. Hence, the length of timeslot 1 is shorter than that of timeslot 2. The usefulness of this special packet is presented in Section 5.5.

However, before transmitting data, a set of rules for aggregating data at the intermediate nodes should also be defined to achieve energy efficiency, and reduce end-to-end delay. The data aggregation policy for the proposed WSN management framework is presented as follows.

The proposed data aggregation scheme is dynamic since the aggregation functions change automatically or by end user based on the application types, and requirements. For instance, if data sensing time of sensors was initially set to a specific time of a day but users need a large number of data for a specific period of time of the day for analysis and achieving more precision, e.g., precision agriculture, data sensing interval should be changed. As soon as the data sensing time or interval is changed, data aggregation policy will also be changed or adjusted accordingly. Thus, the data sensing period, and length of timeslots can be set and changed automatically or by users to get the desired data accuracy that it also reduces the end-to-end data transmission delay.

In some applications, sensor's data only at the level L_i or lower levels that exceed some threshold value (e.g., temperature exceeds 500 degree Celsius) should be reported to BS. Hence, sensor nodes at the level L_{i-1} and upper levels should stop aggregating data. A threshold value for the events of interest is subscribed by BS to the relay nodes. If the aggregating nodes receive any data from the transmitting node that is greater than the threshold value they just forward it to the next hop without any aggregation. Otherwise, aggregating nodes aggregate data based on the selected aggregation functions such as MAX, MIN, and AVERAGE.

The length of timeslots at different levels is also adjusted based on the data types and aggregation functions. In normal aggregation policy, the length of timeslot of sensor nodes at the lowest level is the minimum that is incremented (by a constant) at the upper levels. However, for real-time and time critical emergency applications that stop intermediate nodes to aggregate data, timeslots for nodes at different levels are almost equal to reduce end-to-end data transmission delay. Moreover, in the proposed aggregation scheme, BS can divide a user's query into its fundamental components and distribute the components to nodes at different levels. Nodes gather data for these individual components and aggregate, that result in data reduction and transmit resulted data back to BS. Pseudo-code of this simple aggregation method is presented in Algorithm 2.

7.2 Performance Analysis

We also assume that the data size of sensors at level $L_k = n_{data}$, and data size at aggregating sensor nodes increases by a constant α after aggregation until data arrive at BS.

7.2.1 Aggregated Data Size Exceeds Buffer Space

Let the number of timeslots in a frame $= L_k$, where the average number of active nodes at each level, $\beta \geq 1$. β is calculated during the zone creation, and active nodes selection phases.

Therefore, data size at aggregation nodes at level i is represented by Equation 7.1.

$$(n_{data})_i = \alpha \times \beta \times (L_k - L_i) \times n_{data} \tag{7.1}$$

In Equation 7.1, n_{data} is the size of data sensed by sensors at the lowest level, L_k and $1 \leq i \leq k$.

Algorithm 2 Data Aggregation

BS distributes query to nodes at different levels and/or subscribes for events of interest

if $AG = 0$ (no aggregation) **then**

$AggregationLevel \leftarrow L_i$

No aggregation starting at the level L_i to upper level until data reach BS

$EndtoEndDelay \leftarrow \alpha \times t_i$, where $1 \leq i \leq k$, timeslot t_i at each level is almost equal

Calculate E_{Tx}, transmission energy consumptions

else if $AG = 1$ (aggregation) **then**

if $query = \sum_{i=1}^{n}(query_i)$ **then**

BS divides $query$ into n fundamental components and distribute

else

Each node collects query data

end if

Forward data through aggregating nodes, and aggregates based on $MAX, MIN, andAVG$ functions

Calculate $EndtoEndDelay$ and E_{Tx}

end if

Number of frames that constitute a round is calculated by estimating α (a constant at which data packet size is increased after aggregation, $\alpha = 1$ if the node is not aggregating) so that the buffer of the aggregation node at level 1 does not overflow. Otherwise, a new round would be initiated.

7.2.2 Energy Consumptions

Let us assume that each level has the same number of zones $= \gamma$.

The size of data packet increases at a constant $= \alpha$.

The average number of active nodes at each level, $\leq \beta$

The size of data that is received at BS can be represented by the inequality of Equation 5.10,

$$(n_{data})_{BS} \leq \sum_{i=1}^{k} (\alpha)^i \times \beta \times n_{data} \tag{7.2}$$

Now, we use the energy model that is presented in Equation 5.2 to estimate the energy consumptions, E_{Tx} for transmitting data in the network, which is as follows.

$$E_{Tx} \approx \sum_{i=k}^{i \geq 1} \sum_{j=1}^{\beta} (\alpha)^{k-i+1} \times (n_{data} \times \epsilon_{data} + n_{data} \times (nodeDist[j,i][l,i+1])^2 \times \epsilon_{air}) \tag{7.3}$$

In Equation 5.10, $nodeDist[j,i][l,i+1]$ is the distance from node j at the level i to the neighboring active node l (with which j has a communication path) at the level $i+1$. Similarly, using the energy model that is presented in Equation 5.3, we can estimate the energy consumptions of data reception.

7.2.3 Data Transmission Delay

Let the duration of a timeslot for active nodes at the lowest level$(l_k) = t_l$.

The end-to-end delay, i.e., the time required to reach the aggregated data at BS can be represented by Equation 7.4.

$$delay \geq \sum_{i=1}^{k} (\eta)^{i} \times t_{l} \tag{7.4}$$

In Equation 7.4, η is a constant by which timeslot at each level increases.

7.3 Simulation Setup and Results

We simulate the proposed data aggregation approach to evaluate its performance. Table 7.1 shows the main network parameters, and their respective values that are used in the simulation.

Table 7.1: Simulation parameters, and their respective values

Parameter	**Value**
Network Size	100m X 100m
Number of Nodes	Maximum 100
Number of Zones	Maximum 8
Base station position	55m $\times$ 101m
Data packet size	256 bits
Transmission Energy Consumptions	50 $nJoule/bit$
Energy Consumption in free space or air	0.01 $nJoule/bit/m^2$
Initial Node Energy	2 Joule

We run the simulation for a fixed number of zones, and nodes varying the number of round. We set the number of zones, and nodes to 4 and 40, respectively, and place BS outside all zones at the coordinate (55, 101). We measure the performance of the proposed data aggregation approach in terms of network energy consumptions, end-to-end data transmission delay, and network lifetime by varying the number of rounds.

The network energy consumptions is defined as the energy consumptions of all sensor nodes at different levels for transmitting, receiving, and aggregating data. Network lifetime is defined as the remaining energy of the network after a certain number of round that reflects how long the network works. End-to-end data transmission delay is the total duration of timeslots at different levels of the network.

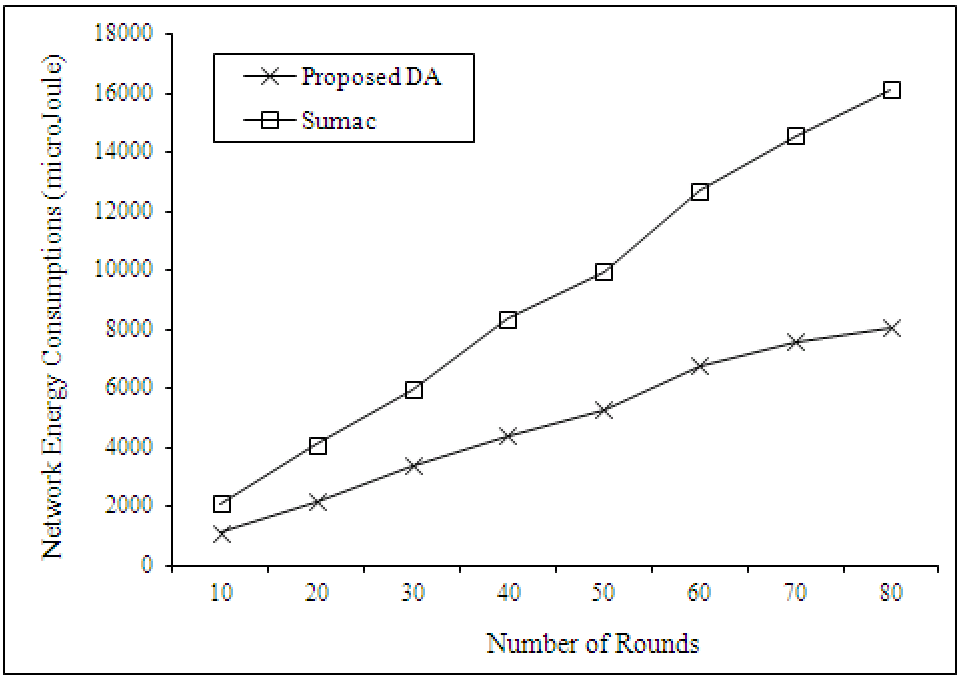

Figure 7.2: Network energy consumptions of over a number of rounds

Figure 7.2 illustrates that the energy consumptions of the proposed data aggregation approach is less than that of the existing SUMAC protocol. Figure 7.3 demonstrates the end-to-end delay for transmitting data from nodes at the lowest level to BS over a number of rounds. It is observed that end-to-end delay in SUMAC is much more than that of the proposed data aggregation approach.

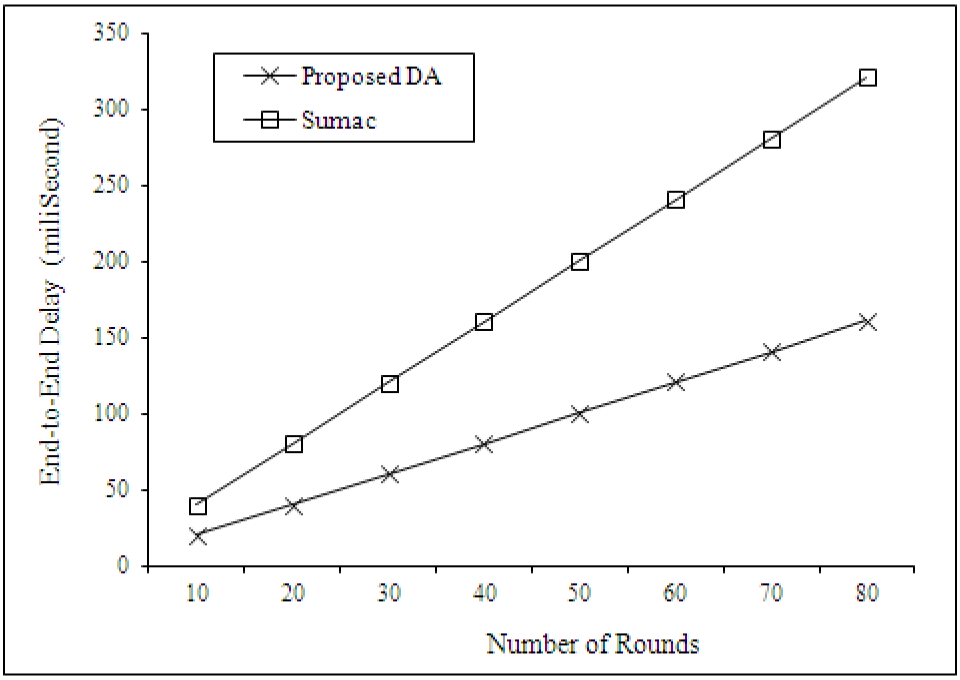

Figure 7.3: End-to-end delay over a number of rounds

Figure 7.4 demonstrates the network lifetime in terms of the remaining energy of the network. We find that the energy dissipation in the SUMAC protocol over a number of rounds is much more than that in the proposed approach. Hence, the SUMAC is expected to have less network lifetime than that of the proposed aggregation method.

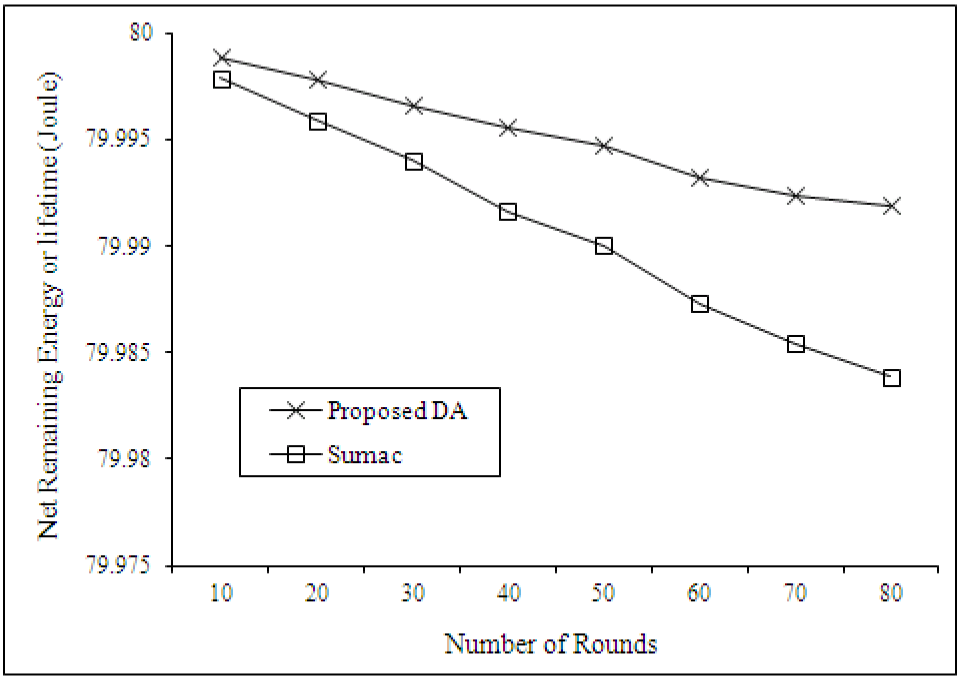

Figure 7.4: Network lifetime over a number of rounds

We also perform student's t-test to investigate whether both systems are really same in terms of network lifetime, energy consumptions, and end-to-end delay. In all cases, we find the $p - value < 0.05$ at the 95% confidence level. Hence, two systems are different, and our proposed data aggregation approach is expected to provide better performance. Possible reasons for these performance differences are explained in the following Section 7.4.

7.4 Discussion

In SUMAC, and most other hierarchical data aggregation schemes, nodes at a level wait for data from the nodes at the lower levels that results in higher end-to-end data transmission delay. Moreover, a large number of aggregating or active nodes at each level results in data redundancy, and processing overhead. The existing schemes claim high data accuracy by using a large number of sensor nodes. However, the proposed scheme ensures data accuracy by selecting an appropriate number of aggregating nodes so that the probability of having sensing hole in the network is very low. Moreover, using the TDMA scheme, nodes sense, and aggregate data at its allocated timeslot. They do not need to wait for data from the lower level nodes that results in lower end-to-end delay. A minimum number of active nodes that covers the whole network area are also expected to provide lower processing overhead. Since most nodes remain idle while only a few active nodes receive, and aggregate data, the total network energy consumptions are expected to be lower as compared to other approaches. On the other hand, a large number of aggregation switching occurs in SUMAC based on the events of interest at different levels, which may degrade the network performance. In the proposed data aggregation scheme, users can control aggregation based on data types, application requirements, and aggregation functions. Hence, the number of aggregation switching between different levels of the network in the proposed approach is lower as compared to SUMAC.

Moreover, the proposed data aggregation approach reduces data or information redundancy to some extent since we select the minimum number of active or aggregating nodes that reduces the sensing overlap and increases the probability of having no sensing hole. Thus, the data aggregation approach reduces aggregating redundant data within the same area though it cannot reduce redundant data sensing in different areas. In addition, sensors do not transmit data packets at regular interval; they only transmit data whenever they sense the subscribed data events. Table 7.2 presents comparative results of the proposed data aggregation approach, SUMAC and some other approaches.

Table 7.2: Comparison of proposed data aggregation with existing approaches

Features	FEDA	CLUDDA	SUMAC	Proposed
Uses clusters	X	√	X	√
Dynamic Aggregation	√	√	√	√
Static Aggregation	X	X	√	√
Fault Tolerance	√	X	X	√
Reduce data Redundancy selecting a few active nodes	X	X	X	√
Uses alternate nodes for fault tolerance	X	X	X	√
Event of interest changes aggregation level automatically	X	X	√	X
Single point of failure	√	X	√	X
Use variable length timeslots	X	X	X	√

7.5 Summary

In this chapter, we presented the proposed dynamic data aggregation approach. Experimental results showed that the proposed data aggregation approach is more energy efficient than that of SUMAC - a standard data aggregation technique for a large scale Wireless Sensor Network (WSN). The proposed data aggregation technique is very useful for real-time applications since it provides fast data delivery to the base station (BS) using TDMA scheme. In addition, the proposed approach provides fault tolerance by allowing alternative paths in case of the failure of a node on the data transmission path. Though we compare the performance of the proposed data aggregation technique only with the SUMAC data aggregation approach we plan to compare the performance of our proposed technique against other fault tolerant data aggregation techniques and consider other factors to achieve better performance and reliability. In the next chapter, we will evaluate the performance of our proposed integrated WSN management framework that comprises the proposed sensors localization, routing, data scheduling and data aggregation approaches.

Chapter 8: Framework Evaluation

In Chapters 4 - 7, we presented the proposed localization, routing, data scheduling, and data aggregation approaches with their performance analysis and evaluation. We have efficiently integrated these approaches into a single framework that can be used as a complete network management solution for a large scale Wireless Sensor Network (WSN). Then, we evaluate the performance of this framework.

This chapter is organized as follows. Section 8.1 presents the performance evaluation of the proposed Range-free and Energy efficient Localization using Mobile Anchor (RELMA) in the integrated framework. Similarly, the performance of the proposed data aggregation approach in the integrated framework is evaluated in Section 8.2. Finally, we replicate the integrated framework by replacing its routing, and localization approaches with the corresponding approaches of an existing WSN management solution [93], and compare this new framework with our proposed framework in Section 8.3.

8.1 RELMA: Localization Approach

We evaluate the performance of our proposed localization approach, RELMA in the integrated WSN management framework in terms of localization error, and energy consumptions and compare it against the existing NBLS [66] localization approach. For this comparison purpose, we form a new framework by replacing RELMA with NBLS in the proposed framework and then compare the new framework with the proposed framework. Table 4.1 of Chapter 4 shows the simulation parameters, and their respective values that we use in this simulation.

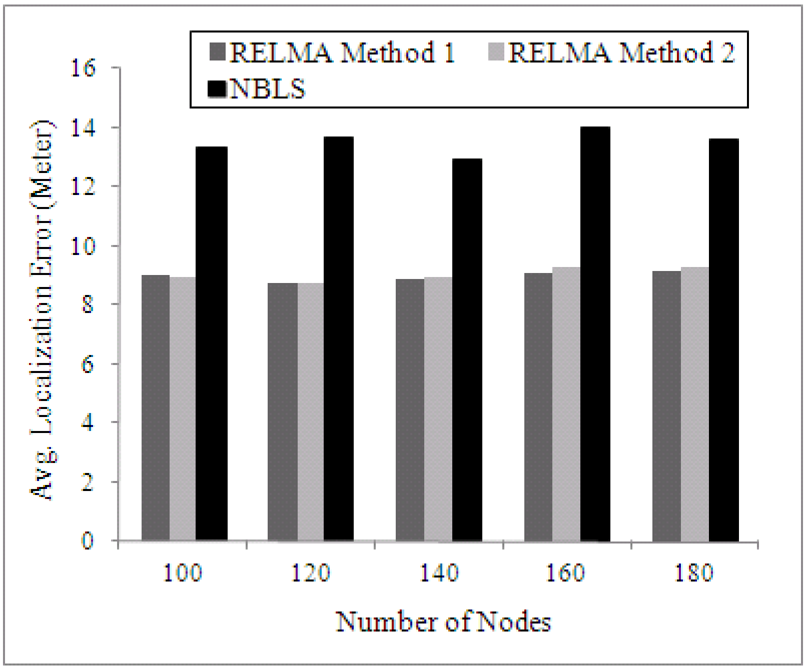

Figure 8.1: Localization error of RELMA Method 1, Method 2, and NBLS in the integrated Framework

We use randomly connected Unit Disk Graphs (UDGs) on an area of 100×100 meters as a network simulation model. We vary the number of nodes between 100, and 180. We set the number of zones, transmission range (R_c), and sensing range (R_s) to 4 zones, 30 meters (R_c), and 15 meters (R_s), respectively. We already mentioned in Chapter 4 that we use a few mobile anchor nodes for sensors localization that are attached to vehicles to move around the network. For the small network area in this simulation, we use a single mobile anchor node, which moves at the velocity 3 meters/second.

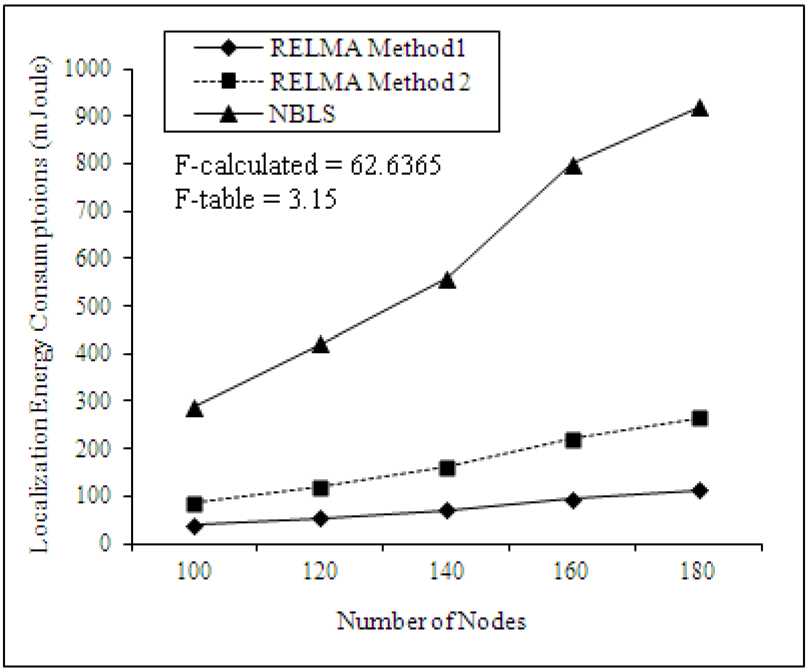

Figure 8.2: Localization energy consumptions of RELMA Method 1, Method 2, and NBLS in the integrated framework

Figures 8.1 and 8.2 illustrate that both RELMA Method 1, and Method 2 outperform the existing NBLS approach in the integrated framework in terms of average localization error, and localization energy consumptions. This is because the existing NBLS localization approach is very complex, and works in three phases. In the first phase, it localizes nodes using DV-hop approach. These initial localized nodes are known as unexamined nodes. In the second phase, the unexamined nodes, which have localization error less than an error threshold are selected as quasi-anchor nodes. These quasi-anchor nodes are used in the third phase to localize all other unexamined nodes using neighbor information based approach. Thus, a large number of messages are transmitted in NBLS approach. Moreover, in NBLS approach, static anchor nodes use communication range, R_c to transmit location information to un-localized nodes, whereas the proposed RELMA localization approach uses sensing range, R_s. Since R_s is much shorter than R_c, RELMA contributes to lower localization energy consumptions and error than that of NBLS. Figure 8.2 also shows the F-values (F-calculated = 62.6365, F-table = 3.15) using ANOVA test at 95% confidence level, which also validates our claim that all approaches are different in terms of localization energy consumptions.

Figure 8.1 also demonstrates that both RELMA Method 1, and Method 2 are identical in terms of the average localization error, which is also validated by student's t-test. However, localization energy consumptions for RELMA Method 1 is less than that of RELMA Method 2 (Figure 8.2). This is because a large number of messages are transmitted in RELMA Method 2 for determining the neighboring set (NS), and neighboring distance (ND) between anchor and un-localized nodes, which is not the case for RELMA Method 1.

8.2 Proposed Data Aggregation

In this section, we evaluate the performance of the proposed data aggregation approach in the integrated framework and compare it against the existing SUMAC data aggregation approach in terms of network energy consumptions, end-to-end data transmission delay, and network lifetime varying both the number of rounds, and zones. For this comparison purpose, we form a new framework by replacing the proposed data aggregation approach with SUMAC in the integrated framework. Then we compare the new framework to the proposed integrated framework. We use the same simulation setup as is presented in Section 8.1.

8.2.1 Varying Number of Rounds

We vary the number of rounds between 10000, and 45000. We set the number of nodes, zones, and sensing range (R_s) to 100 nodes, 4 zones, and 15 meters (R_s), respectively.

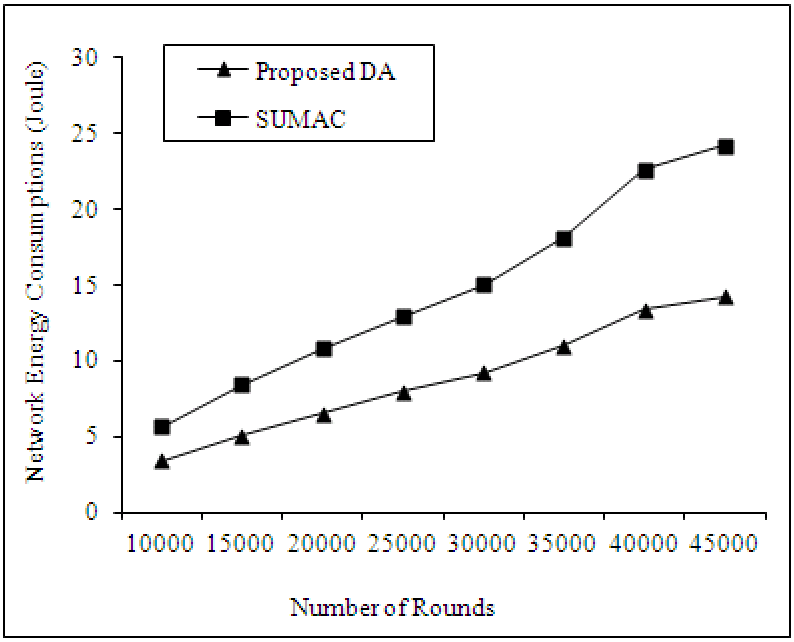

Figure 8.3: Network energy consumptions of the proposed data aggregation approach and SUMAC in the integrated framework over a number of rounds

Figure 8.3 demonstrates that the proposed data aggregation approach has better performance than SUMAC in terms of network energy consumptions over a number of rounds. This is because the proposed data aggregation approach selects a minimum number of aggregating nodes. Moreover, special packets that are transmitted to achieve fault tolerance consume less energy.

Figure 8.4 illustrates that the proposed data aggregation approach has lower end-to-end data transmission delay as compared to the existing SUMAC approach over a number of rounds. This is because the proposed data aggregation approach uses variable length timeslots at different levels of the virtual hierarchy using TDMA scheme. Moreover, small sized special packets take less time to be transmitted to the BS.

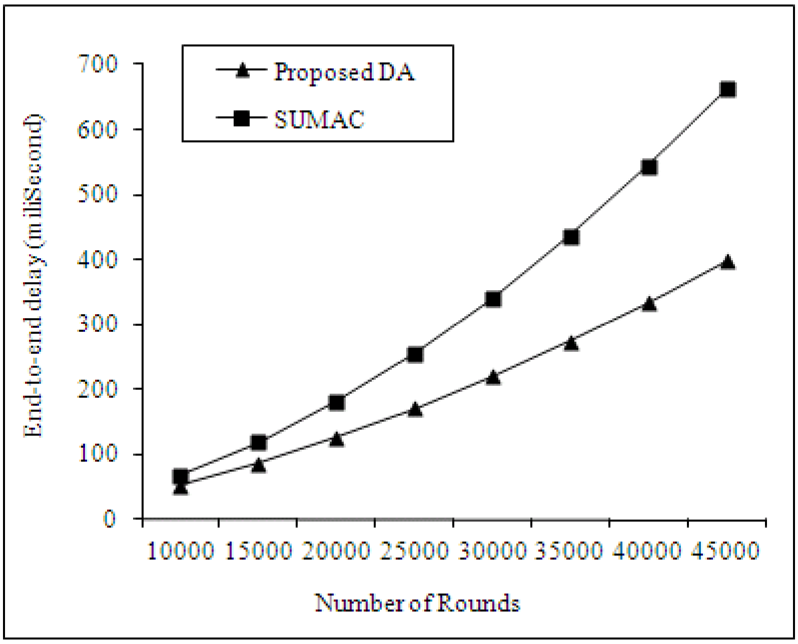

Figure 8.4: End-to-end data transmission delay of the proposed data aggregation approach and SUMAC in the integrated framework over a number of rounds

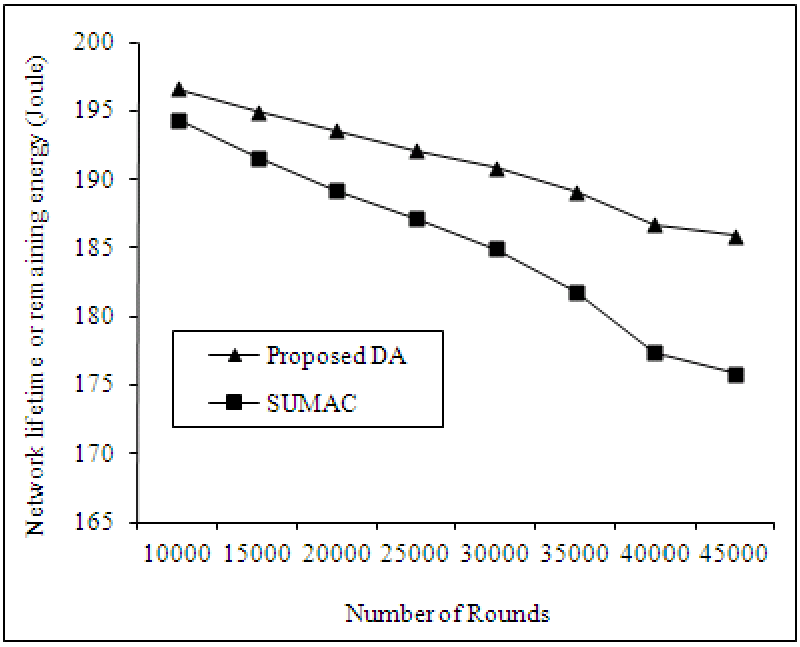

Figure 8.5: Network lifetime of the proposed data aggregation approach and SUMAC in the integrated framework over a number of rounds

Similarly, Figure 8.5 demonstrates that the proposed data aggregation approach outperforms the existing SUMAC approach in terms of network lifetime over a number of rounds. One of the possible reasons is that SUMAC requires a large number of aggregations switching among different levels of the network. This requires a large number of message transmissions and consumes much network energy, which is not the case for the proposed data aggregation approach.

8.2.2 Varying Number of Zones

We also evaluate the performance of the proposed framework varying the number of zones between 4, and 10. We set the number of nodes, rounds, and sensing range (R_s) to 120 nodes, 10000, and 15 meters (R_s), respectively.

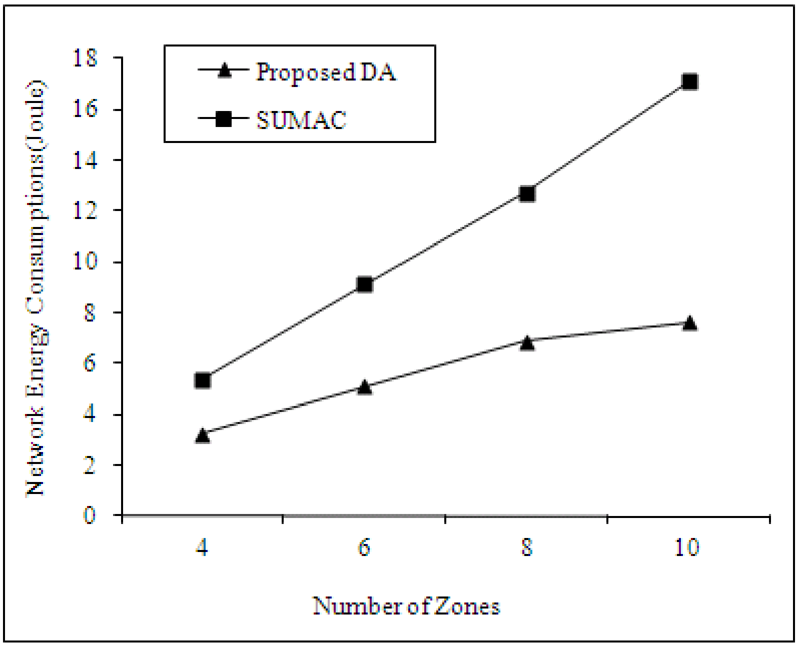

Figure 8.6: Network energy consumptions of the proposed data aggregation approach and SUMAC in the integrated framework over a number of zones

Figure 8.6 illustrates that the proposed data aggregation approach has much lower network energy consumptions as compared to the existing SUMAC data aggregation approach. Network energy consumptions increase for increasing the number of zones because the number of active nodes increases for increasing the number of zones even if the total number of nodes in the network is fixed. We also observe that the increase of network energy consumptions between two consecutive zones (e.g., between zones 8 and 10) in the proposed data aggregation approach is also much lower than that in SUMAC data aggregation approach. This is because energy consumptions are directly proportional to the square of a distance (d^2). Increasing the number of zones in the fixed network area also increases the number of virtual levels and reduces the distance between two active nodes on the shortest data transmission path in the proposed approach.

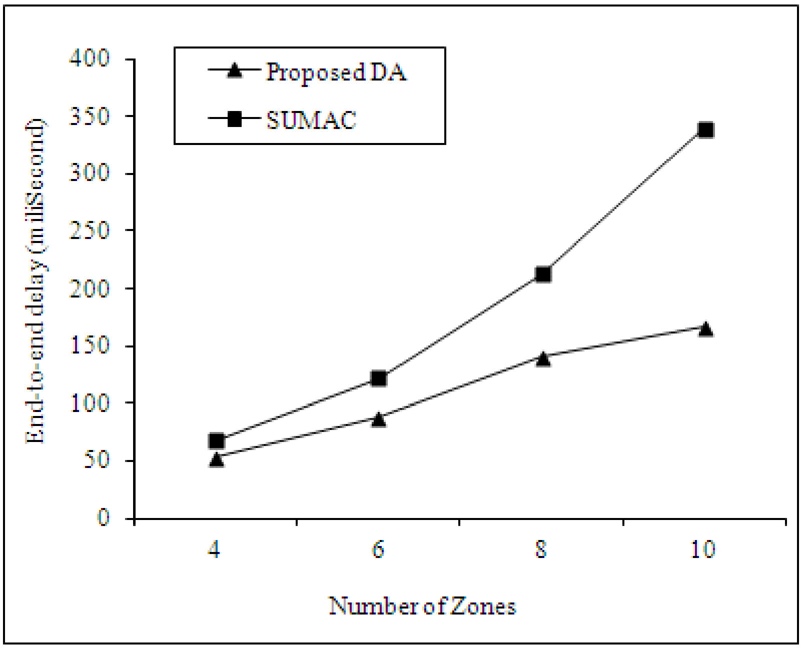

Figure 8.7: End-to-end data transmission delay of the proposed data aggregation approach and SUMAC in the integrated framework over a number of zones

Figure 8.7 demonstrates the end-to-end data transmission delay in the proposed data aggregation approach is much lower than than in the SUMAC approach because the proposed data aggregation uses variable length timeslots at different levels of the virtual data aggregation tree. Moreover, the proposed approach uses the shortest data transmission paths from the source node to the BS, whereas data transmission delay is proportional to distance.

8.3 Proposed Framework

To best of my knowledge, no integrated framework comprising sensors localization, routing, data scheduling, and data aggregation approaches exists in the literature. However, we find an approach [93] that comprises sensors localization, and routing protocols. Thus, we build a new Wireless Sensor Network (WSN) management framework by replacing the localization, and routing approaches in the proposed framework with the corresponding approaches in [93]. Then, we compare the performance of the proposed framework with this new framework in terms of localization energy consumptions, localization error, and network energy consumptions.

In the following subsections, we briefly present the existing approach that comprises localization, and routing protocols.

8.3.1 Existing Approach

Existing WSN management approach [93] has two components: sensors localization, and routing protocol. In the localization approach, the sink node acts as an origin that is located at the co-ordinate (0, 0). One of the neighboring nodes, A of sink is considered to be located on the positive side of x-axis at the co-ordinate $(a, 0)$. Then, the sink node collects all neighboring tables that contain information of distances among sensor nodes. Using this information, sink selects a node, B, which is the neighbor of both sink and A. Authors assume that node B is at distance b_1 from sink and distance b_2 from node A, and node B is located at the positive side of y-axis. Once the locations of sink, A and B are known, other nodes that receive the location information of these three nodes can localize themselves using triangulations. In this approach, whenever a node, C is localized it is considered as an anchor node. Thus, if there is any location error in C, this error will be propagated to localize other un-localized nodes. This results a large average localization error.

In the routing protocol of this existing approach, each node keeps an attribute, radio depth d_r, (i.e., distance from the node to sink in terms of radio range). For instance, the sink has $d_r = 0$, and all nodes within the radio range of sink has $d_r = 1$. Whenever a node, A checks its neighbor table, and finds the node m with the lowest radio depth, d_m, A sets its $d_r(A)$ as $d_r(A) = d_m + 1$. In routing towards sink, each node selects the upstream node with radio depth $d_r - 1$ as the next hop. However, this selection based on the least hop in terms of radio range does not guarantee the shortest path in Euclidean distance. This routing protocol also considers the remaining energy of a node along with its d_r to balance energy network consumptions. In the following subsection, we present simulation results.

8.3.2 Simulation Results

Existing approach measures the performance of its localization approach in terms of normalized root mean square error (NRMSE) by varying the signal-to-noise ratio, and routing protocol in terms of network lifetime by varying the radio range. However, we use different performance metrics in all our proposed WSN management components, and framework. Thus, we also simulate the existing approach to compare with the proposed framework rather than using the results from the paper [93].

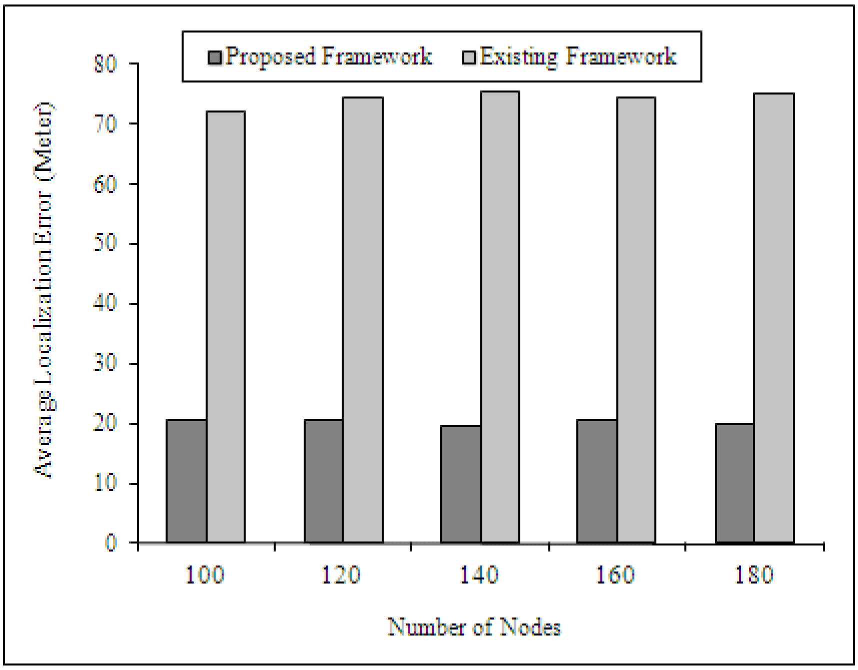

Figure 8.8: Localization error varying the number of nodes

We use the same simulation setup as is presented in Section 8.1. We vary the number of nodes between 100, and 160. We set the number of zones, rounds, and sensing range to 4 zones, 100, and 28 meters (R_s), respectively. Figure 8.8 demonstrates that the average localization error of the existing (or newly formed) management framework is much higher than that of the proposed framework. This is because the existing approach considers a newly localized node as an anchor node. Thus, error in localizing a node will be propagated and results in a large localization error. We illustrate the average localization error, where, $avgLocError = \frac{totalError}{numberofnodes}$. Thus, the error remains almost same even if we increase the number of nodes.

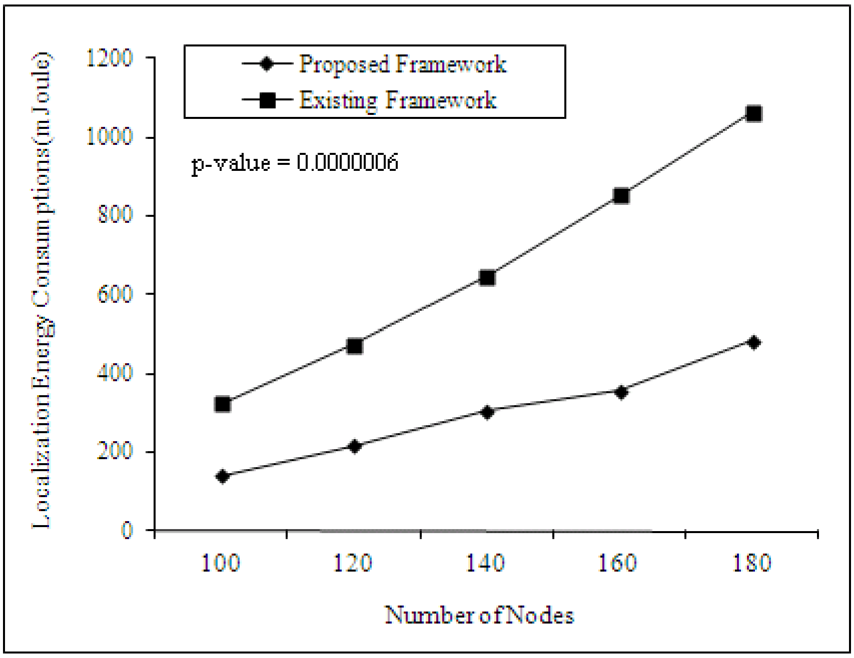

Figure 8.9: Localization energy consumptions varying the number of nodes

Figure 8.9 illustrates that the localization energy consumptions of the proposed framework is much lower than that of the existing localization approach in [93]. This is because each node consumes much energy to compute its neighboring table (with distance information to each other) in the existing localization approach whereas, in the proposed approach, anchor nodes transmits their location information to the un-localized nodes, which are at the smaller sensing range apart. Figure 8.9 also shows that the p-value of statistical analysis (student's t-test) at 95% confidence level is equal to 0.0000006, which validates the claim.

Initially, we estimate the localization error in both RELMA Method 1 and Method 2 using the approach stated in Sections 4.1.1 and 4.1.2. To estimate the localization error in the proposed integrated framework, we consider the lower localization error that is found either using RELMA Method 1 or RELMA Method 2 as the localization error of a node P. We have shown in Chapter 4 through simulation results that the localization error of sensor nodes are almost the same both using the RELMA Method

1 and Method 2. Thus, the localization error in the integrated framework will also be the same either using the RELMA Method 1 or RELMA Method 2. However, we demonstrate that the localization energy consumptions in RELMA Method 1 is much lower than that of the RELMA Method 2 (have already explained the reasons). Thus, we consider the localization energy consumptions in RELMA Method 1 as the localization energy consumptions in the proposed integrated framework.

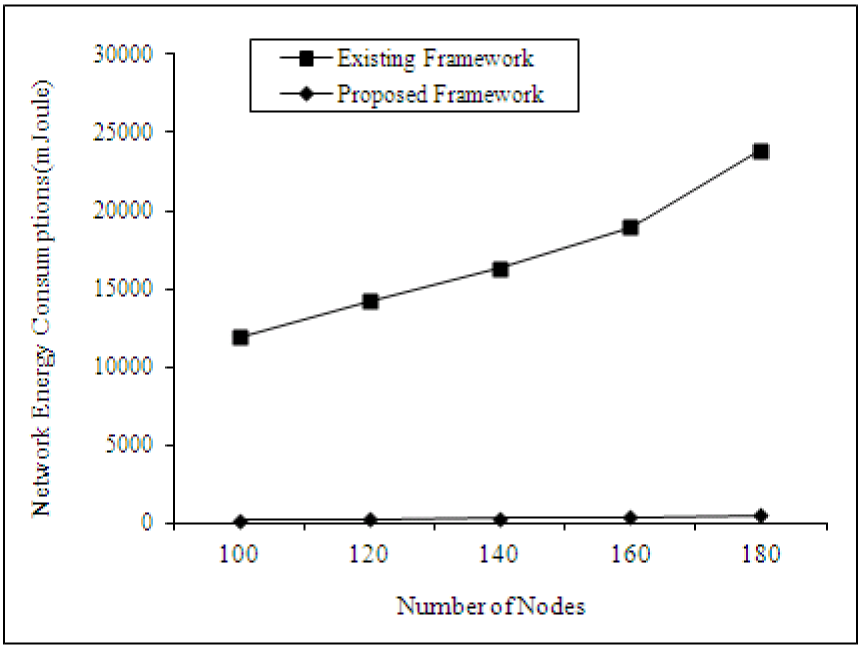

Figure 8.10: Network energy consumptions varying the number of nodes

Figure 8.10 demonstrates that the network energy consumptions (that contains energy consumptions for localization, and routing) in the proposed approach is much lower than that in the existing (or newly formed) approach for varying the number of nodes. This is because each node broadcasts its radio depth to neighboring nodes, which consumes much energy in the existing routing protocol. Moreover, in the existing routing protocol, a sensor node transmits data to its neighboring node with the lowest radio depth. However, a neighboring node with the least radio depth does not guarantee the shortest euclidian distance.

We also observe that the difference in the network energy consumptions between the existing, and proposed frameworks is much more in Figure 8.10 than this difference in Figure 8.9. This is because the energy consumptions of the existing framework for data routing is also much higher than that of the proposed framework. Thus, the difference for the network energy consumptions between the existing, and proposed framework increases since the network energy consumptions includes energy consumptions for both localization, and routing.

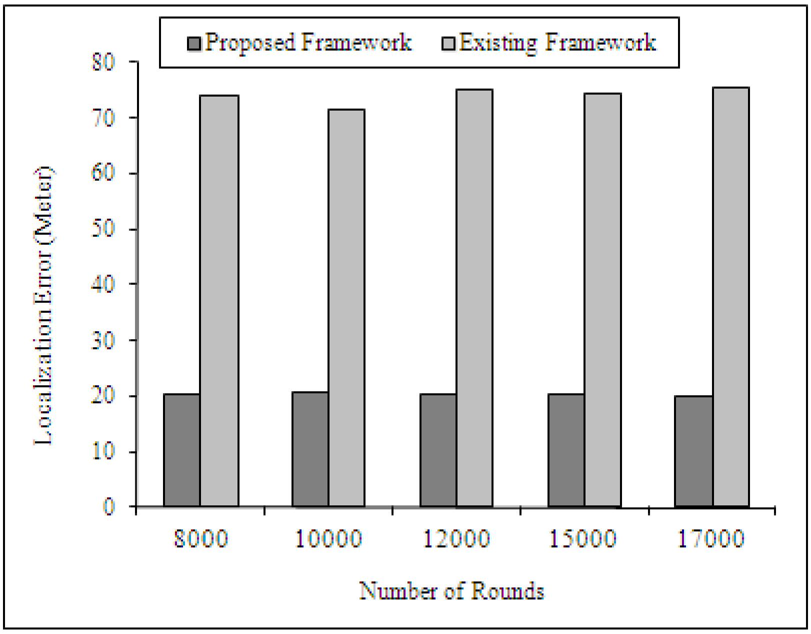

Figure 8.11: Localization error varying the number of rounds

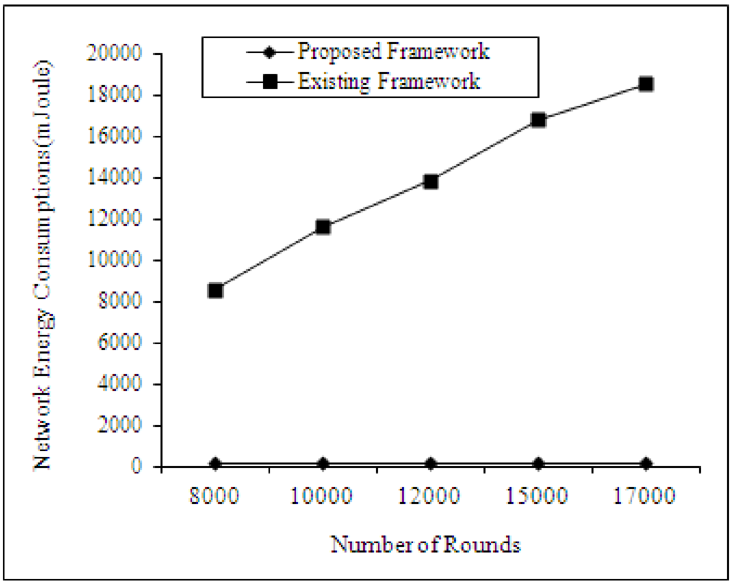

Figure 8.12: Network energy consumptions varying the number of rounds

We also evaluate the performance of the proposed framework by varying the number of rounds while we set the number of nodes, and zones, to 100, and 4, respectively. Figures 8.11, and 8.12 demonstrate the performance results in terms of network energy consumptions, and end-to-end delay, respectively. In both cases, the proposed framework outperforms the existing framework. The possible reasons for these performance differences have already been explained.

8.4 Summary

In this chapter, we evaluated the performance of the proposed integrated Wireless Sensor Network (WSN) management, and control framework. Simulation results show that the proposed framework outperforms an existing approach that comprises only localization, and routing protocols in terms of localization energy consumptions, localization error, network energy consumptions, and network lifetime for varying number of nodes, and rounds. Thus, this framework can be used in a large scale WSN to prolong the network life time, and achieve reliability. We expect this framework can be used as a basis for developing efficient and large scale WSN management frameworks in future. In the next chapter, we conclude this thesis, and present ideas for future work.

Chapter 9: Conclusion and Future Work

In this chapter, we present a summary of the contributions of this research, and discuss ideas for future research directions.

9.1 Summary

In this PhD thesis, we investigated a large scale multi-modal Wireless Sensor Network (WSN) architecture that consists of three networks: WSN, Wi-Fi, and WiMAX/GPRS, and design an efficient, and integrated network management framework comprising of sensors localization, packet scheduling, routing protocol, and data aggregation approaches for the WSN plane of this architecture. Initially, we introduce an efficient approach for each of these network management components such as we design a range free, accurate, and energy efficient localization approach. Then, we design efficient packet scheduling, routing, and data aggregation approaches. We analysis, and evaluate the performance of each of the network management components such as localization, routing, data scheduling, and data aggregation and compare them with existing approaches. Experimental results show that the proposed approaches outperform existing approaches in terms of localization energy consumptions, average localization error, network energy consumptions, end-to-end data transmission delay, and network lifetime. We also validate the performance of the proposed approaches through statistical analysis (student't t-test).

We have integrated the network management components so that they efficiently work together for a large scale WSN in terms of energy consumptions, localization error, end-to-end data transmission delay, and packet delivery ratio. We form a new framework by replacing the proposed localization approach, RELMA with the existing NBLS localization approach in the integrated framework and then compare the performance of the proposed framework with the new framework. This actually compares the performance of the RELMA localization approach with NBLS localization approach in the integrated framework. Similarly, we compare the proposed data aggregation, and routing protocol with existing approaches in the integrated framework. In all cases, the proposed approaches have better performance than their existing counterparts.

To best of our knowledge, there is no existing WSN framework that comprises all these network management components. Thus, we again form a new framework by replacing both localization and routing protocol of the integrated framework with the corresponding components in an existing solution [93] that only comprises localization scheme, and routing protocol. From the experimental results we find that the proposed framework outperforms the existing framework in terms of localization energy consumptions, localization error, routing energy consumptions, and network energy consumptions. In the following section, we present the contributions of this thesis.

9.2 Contributions

The contributions of this thesis are as follows:

- A Range-free Energy efficient, Localization technique using Mobile Anchor (RELMA) that has lower localization error, and energy consumptions as compared to the existing approaches. In RELMA, anchor nodes transmit their locations to un-localized nodes, which are at the sensing range (short) apart from the anchor node. Thus, the transmission energy consumptions of RELMA will be much less than the existing approaches, which use communication range. Since the sensing range is much shorter than the communication range, the region of the intersected sensing circles of three anchor node (inside which the un-localized node resides) will be very small. Thus, the localization error of RELMA will be lower than that of the existing approaches.

- A Dynamic Multilevel Priority (DMP) packet scheduling scheme that provides faster data delivery of emergency real-time data because real-time emergency data have the highest priority, and can preempt other types of data. DMP achieves fairness by allowing the lowest priority non-real-time local data that is sensed at the current node x to transmit after a certain number of rounds if real-time data is not available but the higher priority non-real-time remote data (non-real-time data that x receives from the lower level nodes) are available.

- A zone-based routing protocol, and a dynamic data aggregation approach, which are energy efficient. These approaches use a minimum number of active/aggregating nodes. Thus, they reduce redundant data sensing. These approaches use small sized special packets. Transmitting these small sized packet consumes less energy and results in lower end-to-end delay (i.e., requires less data transmission time). Moreover, these approaches use TDMA scheme having variable length timeslots and provide fast data delivery. The proposed routing protocol also has the following characteristics:

  – It is fault tolerant since it detects the failure of sensor nodes using special packets and retransmits data through alternative paths. A node x transmits special packets to the nodes at upper level whenever a x does not have any subscribed event to transmit. Otherwise, if x does not send any packet the upper level nodes or BS cannot understand whether x has failed or not.

– It balances individual nodes energy consumptions by initiating a network setup phase after a certain number of rounds, which is dynamically calculated based on the network's energy status. Each active node on a data transmission path transmits data containing its remaining energy information to the base station (BS). Thus, BS selects an alternative path or node if the remaining energy of an active node on a data transmission path goes below a threshold value. Thus, the network lifetime is prolonged.

9.3 Future Work

For future work, we plan to compare the performance of each component of the proposed framework with other existing approaches and consider other factors such as sensors radio irregularity and data transmission deadline to improve the performance of the proposed framework. We would also perform experiments in testbed with real sensors, Wi-Fi and WiMAX nodes and measure the effectiveness of the proposed framework for sensor plane. Based on the outcomes we might need to consider designing the network management tools for other layers (Wi-Fi, and WiMAX) for reducing end-to-end data transmission delay, which is a focus of our future research. Other important design issues that we need to investigate is the number of Wi-Fi enabled BS, and the placement of these BS(s). In the proposed framework, we consider a single BS for each sensor cloud, which might result a single point of failure and incur a large number traffic and congestion. Moreover, a single BS for each sensor cloud does not provide scalability when the number of sensor nodes increases. Thus, we plan to use multiple BSs in the framework so that (i) sensor nodes near the BS can select the shortest distant BS to further reduce energy consumptions, and (ii) network is still operable in case of the failure of a BS.

In the proposed framework, localization, routing protocols, packet scheduling, and data aggregation approaches are designed for static sensor networks. We also plan to extend our work to develop a framework that works for mobile sensor nodes and networks. Thus, this framework can be used for a large number of applications such as health monitoring, and wildlife monitoring. Moreover, the proposed framework make an assumption that the radio pattern of the sensor network is regular. We also plan to consider irregular radio pattern for designing the network management components of the proposed framework.

Bibliography

[1] A. Ahmed, H. Shi, , and Shang Y. A survey on routing protocols for wireless sensor networks. In *IEEE International Conference on Information Technology: Research and Education (ITRE)*, pages 301 – 305, Aug. 2003.

[2] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, 2005.

[3] B. A. Alsaify and D. R. Thompson. Pendulum: An energy efficient protocol for wireless sensor networks. In *Sensors Applications Symposium (SAS), 2010 IEEE*, pages 273 – 277, 2010.

[4] S. Anandamurugan and C. Venkatesh. SSP: A sleep scheduling protocol for improving the lifetime of wireless sensor networks. In *Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference on*, pages 1–6, 2009.

[5] G. Anastasi, M. Conti, and M. Di Francesco. Extending the lifetime of wireless sensor networks through adaptive sleep. *Industrial Informatics, IEEE Transactions on*, 5(3):351–365, 2009.

[6] M. H. Anisi, J. Rezazadeh, and M. Dehghan. FEDA: Fault-tolerant energy-efficient data aggregation in wireless sensor networks. *Software, Telecommu-*

nications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on, pages 188–192, 2008.

[7] D. Baghyalakshmi, J. Ebenezer, and S. A. V. Satyamurty. Low latency and energy efficient routing protocols for wireless sensor networks. In *Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on*, pages 1 – 6, 2010.

[8] F. Bajaber and A. Irfan. Dynamic/static clustering protocol for wireless sensor network. *Computer Modeling and Simulation, 2008. EMS '08. Second UKSIM European Symposium on*, pages 524 – 529, 2008.

[9] B. A. Bakr and L. Lilien. LEACH-SM: A protocol for extending wireless sensor network lifetime by management of spare nodes. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, page 375, May 2011.

[10] M. Bal, M. Liu, W. Shen, and H. Ghenniwa. Localization in cooperative wireless sensor networks: A review. In *Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on*, pages 438–443, 2009.

[11] I. Banerjee, P. Chanak, B. K. Sikdar, and H. Rahaman. EER: Energy efficient routing in wireless sensor networks. In *Students' Technology Symposium (TechSym), 2011 IEEE*, pages 92 –97, jan. 2011.

[12] P. Banerjee, D. Jacobson, and S. N. Lahiri. Security and performance analysis of a secure clustering protocol for sensor networks. *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*, pages 145 – 152, 2007.

[13] G. Bergmann, M. Molnar, L. Gonczy, and B. Cousin. Optimal period length for the cgs sensor network scheduling algorithm. In *Networking and Services (ICNS), 2010 Sixth International Conference on*, pages 192–199, 2010.

[14] C. Bolin and Z. Zengwei. LLSiWSN: A new range-free localization algorithm for large scale wireless sensor netwoks. In *Business Computing and Global Informatization (BCGIN), 2011 International Conference on*, pages 408 – 411, July 2011.

[15] E. Bulut and I. Korpeoglu. DSSP: A dynamic sleep scheduling protocol for prolonging the lifetime of wireless sensor networks. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 725–730, 2007.

[16] S. Chachra and M. Marefat. Distributed algorithms for sleep scheduling in wireless sensor networks. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3101–3107, 2006.

[17] C. M. Chao and T. Y. Hsiao. Design of structure-free and energy-balanced data aggregation in wireless sensor networks. *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, pages 222–229, 2009.

[18] S. Chatterjea and P. Havinga. A dynamic data aggregation scheme for wireless sensor networks. *Computer*, pages 346–350, 2007.

[19] S. K. Chaurasiya, T. Pal, and S. D. Bit. An enhanced energy-efficient protocol with static clustering for wsn. In *Information Networking (ICOIN), 2011 International Conference on*, pages 58–63, 2011.

[20] V. K. Chaurasiya, R. L. Lavavanshi, S. Verma, G. C. Nandi, and A. K. Srivastava. Localization in wireless sensor networks using directional antenna. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 131–134, 2009.

[21] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, H. Tejeda, and J. Urrutia. Half-space proximal: A new local test for extracting a bounded dilation spanner. In *In SPRINGERVERLAG, ed., OPODIS 05: Proceedings of the 9th International Conference on Principles of Distributed Systems*, 2005.

[22] R. Chen, Z. Zhong, and M. Ni. Cluster based iterative gps-free localization for wireless sensor networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1 – 5, May 2011.

[23] F. Dressler. *Self-Organization in Sensor and Actor Networks*. John Wiley Sons Ltd, 2007.

[24] C. Duan and H. Fan. A distributed energy balance clustering protocol for heterogeneous wireless sensor networks. *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2469–2473, 2007.

[25] Z. Duan, F. Guo, M. Deng, and M. Yu. Shortest path routing protocol for multi-layer mobile wireless sensor networks. In *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 02*, pages 106–110, Washington, DC, USA, 2009. IEEE Computer Society.

[26] A. Durresia, V. Paruchuri, and L. Barolli. Clustering protocol for sensor net-

works. *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, 2:5 pp., 2006.

[27] N. Edalat, W. Xiao, C.K. Tham, E. Keikha, and L. Ong. A price-based adaptive task allocation for wireless sensor network. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 888–893, 2009.

[28] M. Elhawary and Z. J. Haas. Energy-efficient protocol for cooperative networks. *Networking, IEEE/ACM Transactions on*, 19(2):561 –574, april 2011.

[29] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259 – 278, Sept 1969.

[30] A. Garg and M. Hanmandlu. An energy-aware adaptive clustering protocol for sensor networks. *Intelligent Sensing and Information Processing, 2006. ICISIP 2006. Fourth International Conference on*, pages 23–30, 2006.

[31] Soheil Ghiasi, Ankur Srivastava, Xiaojian Yang, and Majid Sarrafzadeh. Optimal energy aware clustering in sensor networks. *Sensors*, 2:258–269, 2002.

[32] L. Gui, T. Val, and A. Wei. Improving localization accuracy using selective 3-anchor dv-hop algorithm. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1 – 5, Sept. 2011.

[33] P. Guo, T. Jiang, Q. Zhang, and K. Zhang. Sleep scheduling for critical event monitoring in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(2):345 – 352, Feb. 2012.

[34] Z. J. Haas and M. R. Pearlman. The performance of query control schemes for the zone routing protocol. *Networking, IEEE/ACM Transactions on*, 9(4):427 –438, aug 2001.

[35] J. He. Cooperative range-free localization for wireless sensor networks. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, pages 1 – 6, Oct. 2011.

[36] T. He, C. Huang, Brian M. Blum, John A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 81 – 95, New York, NY, USA, 2003. ACM.

[37] Y. He and H. Li. A distributed node localization algorithm based on believable factor for wireless sensor network. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1–4, 2009.

[38] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660 – 670, Oct 2002.

[39] X. Hu, D. Ren, H. Wang, Y. Qiu, and C. Jiang. Adaptive clustering algorithm based on energy restriction. In *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, volume 2, pages 949–951, 2011.

[40] B. Huang, F. Hao, H. Zhu, Y. Tanabe, and T. Baba. Low-energy static clustering scheme for wireless sensor network. *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on*, pages 1–4, 2006.

[41] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. *Distributed Com-*

puting Systems, 2002. Proceedings. 22nd International Conference on, pages 457–458, 2002.

[42] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A scalable and robust communication paradigm for sensor networks. In *MOBICOM*, pages 56–67. ACM, 2000.

[43] X. Jian-bo and P. Li. A new intra-cluster node scheduling algorithm of clustering data collection protocol in wsns. *Electronic Computer Technology, 2009 International Conference on*, pages 582–586, 2009.

[44] H. Jiang, S. Jin, and C. Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064 –1071, June 2011.

[45] C. Joo, J.G. Choi, and N. B. Shroff. Delay performance of scheduling with data aggregation in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, 2010.

[46] R. Jurdak, A. Nafaa, and A. Barbirator. Large scale environmental monitoring through integration of sensor and mesh networks. *MDPI Sensors*, 2008.

[47] L. Karim, T. El Salti, and N. Nasser. Routing on mini-gabriel graphs in wireless sensor networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 105 –110, Oct. 2011.

[48] L. Karim, N. Nasser, H. Abdulsalam, and I. Moukadem. An efficient data aggregation approach for large scale wireless sensor networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1 – 6, Dec. 2010.

[49] L. Karim, N. Nasser, and T. El Salti. RELMA: A range free localization approach using mobile anchor node for wireless sensor networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1 –5, Dec. 2010.

[50] L. Karim, N. Nasser, and T. El Salti. Efficient zone-based routing protocol of sensor network in agriculture monitoring systems. In *Communications and Information Technology (ICCIT), 2011 International Conference on*, pages 167 –170, March 2011.

[51] L. Karim, N. Nasser, T. E. Salti, and Q. H. Mahmood. The significant impact of a set of topologies on wireless sensor networks. *Accepted to be published in the EURASIP Journal on Wireless Communications and Networking*, 2011.

[52] L. Karim, N. Nasser, and T. Sheltami. A fault tolerant dynamic clustering protocol of wireless sensor networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –6, 30 2009-dec. 4 2009.

[53] L. Karim, N. Nasser, and Sheltami T. A fault-tolerant energy-efficient clustering protocol of a wireless sensor network. *Wireless Communication and Mobile Computing*, 2011.

[54] L. Karim, N. Nasser, Taleb T., and Alqallaf A. K. An efficient priority packet scheduling algorithm for wireless sensor network. In *Accepted to be published in Communications (ICC), 2012 IEEE International Conference on*, June 2012.

[55] E. Karimi and B. Akbari. Improving video delivery over wireless multimedia sensor networks based on queue priority scheduling. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 – 4, Sept. 2011.

[56] V. Katiyar, N. Chand, G. C. Gautam, and A. Kumar. Improvement in LEACH protocol for large-scale wireless sensor networks. In *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*, pages 1070–1075, 2011.

[57] W. Ke, W. Liqiang, C. Shiyu, and Q. Song. An energy-saving algorithm of wsn based on gabriel graph. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1 – 4, sept. 2009.

[58] O. Khader, A. Willig, and A. Wolisz. Distributed wakeup scheduling scheme for supporting periodic traffic in wsns. In *Wireless Conference, 2009. EW 2009. European*, pages 287–292, 2009.

[59] K. Kim, H. Kim, and Y. Hong. A self localization scheme for mobile wireless sensor networks. In *Computer Sciences and Convergence Information Technology, 2009. ICCIT '09. Fourth International Conference on*, pages 774 – 778, 2009.

[60] Y. Kim, C. Yang, and C. H. Liu. Throughput analysis of randomized sleep scheduling with constrained connectivity in wireless sensor networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6, 2008.

[61] L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575 – 578, 2002.

[62] B. F. La Scala, X. Wang, and B. Moran. Node self-localisation in large scale

sensor networks. In *Information, Decision and Control, 2007. IDC '07*, pages 188–192, 2007.

[63] E. M. Lee, A. Kashif, D. H. Lee, I. T. Kim, and M. S. Park. Location based multi-queue scheduler in wireless sensor network. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 1, pages 551–555, 2010.

[64] S. Lee and K. Kim. Determination of communication range for range-free multi-hop localization in wireless sensor networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1 –4, Aug 2011.

[65] P. A. Levis. TinyOS: An open operating system for wireless sensor networks (invited seminar). In *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, page 63, May 2006.

[66] S. Li, L. Huang, J. Wu, H. Xu, and J. Wang. NBLS: Neighbor-information-based localization system for wireless sensor networks. In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pages 91–94, 2008.

[67] S. Li, D. Qian, Y. Liu, and J. Tong. Cluster-based multi-path data dissemination scheme for large scale wireless sensor networks. *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2715–2718, 2007.

[68] Y. Liang and H. Gao. An energy-efficient clustering algorithm for data gathering and aggregation in sensor networks. *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, pages 3935–3939, 2009.

[69] K. Lin, H. Zhao, Z. Y. Yin, and Y. G. Bi. An adaptive double ring scheduling strategy based on tinyos. *Journal of Northeastern University Natural Science*, 28(7):985 – 988, 2007.

[70] S. Lindsey and C. S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*, 2002.

[71] F. Liu, C. Tsui, and Y. J. Zhang. Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 9(7):2258 – 2267, July 2010.

[72] J. Liu, N. Gu, and S. He. An energy-aware coverage based node scheduling scheme for wireless sensor networks. *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 462–468, 2008.

[73] S. Liu, T. Fevens, and A. E. Abdallah. Hybrid position-based routing algorithms for 3d Mobile Ad Hoc Networks. In *Mobile Ad-hoc and Sensor Networks, 2008. MSN 2008. The 4th International Conference on*, pages 177 – 186, dec. 2008.

[74] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. RAP: a real-time communication architecture for large-scale wireless sensor networks. In *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, pages 55 – 66, 2002.

[75] Q. Lu, M. Bai, W. Zhang, and P. Yanli. A new kind of NDV-hop algorithm in wireless sensor network. In *Network Computing and Information Security (NCIS), 2011 International Conference on*, volume 2, pages 438 – 441, May 2011.

[76] Madhulika, A. Kumar, and S. Varma. Iterative and distributed range-free localization algorithm for wireless sensor networks. In *Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT '09. International*, pages 102–105, 2009.

[77] A. Manjeshwar and D. P. Agrawal. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. *Parallel and Distributed Processing Symposium, International*, 3:30189a, 2001.

[78] A. Manjeshwar and D. P. Agrawal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, IPDPS '02, pages 48–55, Washington, DC, USA, 2002. IEEE Computer Society.

[79] A. Martirosyan and A. Boukerche. Performance evaluation of an energy-aware clustering protocol for wireless sensor networks. *Parallel Processing - Workshops, 2008. ICPP-W '08. International Conference on*, pages 67–72, 2008.

[80] A. Martirosyan, A. Boukerche, and R. W. N. Pazzi. A taxonomy of cluster-based routing protocols for wireless sensor networks. *Parallel Architectures, Algorithms, and Networks, 2008. I-SPAN 2008. International Symposium on*, pages 247–253, 2008.

[81] D. W. Matula and R. R. Sokal. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12:205 – 222, 1980.

[82] L. Meng, H. Zhang, and Y. Zou. A data aggregation transfer protocol based on clustering and data prediction in wireless sensor networks. In *Wireless Commu-*

nications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on, pages 1 –5, sept. 2011.

[83] Q. Meng and Xu H. M. A distributed range-free localization algorithm based on clustering for wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2633 –2636, sept. 2007.

[84] T. Minming, N. Jieru, W. Hu, and L. Xiaowen. A data aggregation model for underground wireless sensor network. *Computer Science and Information Engineering, 2009 WRI World Congress on*, 1:344–348, 2009.

[85] F. Mirian and M. Sabaei. A delay and accuracy sensitive data aggregation structure in wireless sensor networks. In *Information Management and Engineering, 2009. ICIME '09. International Conference on*, pages 231 –235, April 2009.

[86] K. Mizanian, R. Hajisheykhi, M. Baharloo, and A. H. Jahangir. RACE: A real-time scheduling policy and communication architecture for large-scale wireless sensor networks. In *Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual*, pages 458–460, 2009.

[87] K. Mizanian, R. Hajisheykhi, M. Baharloo, and A. H. Jahangir. RACE: A real-time scheduling policy and communication architecture for large-scale wireless sensor networks. In *Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual*, pages 458 – 460, may 2009.

[88] H. Momeni, M. Sharifi, and S. Sedighian. A new approach to task allocation in wireless sensor actor networks. In *Computational Intelligence, Communication*

Systems and Networks, 2009. CICSYN '09. First International Conference on, pages 73–78, 2009.

[89] B. Nazir and H. Hasbullah. Dynamic sleep scheduling for minimizing delay in wireless sensor network. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, pages 1 – 5, April 2011.

[90] P. Neamatollahi, H. Taheri, M. Naghibzadeh, and M. Yaghmaee. A hybrid clustering approach for prolonging lifetime in wireless sensor networks. In *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*, pages 170 –174, feb. 2011.

[91] P. Nie and B. Li. A cluster-based data aggregation architecture in wsn for structural health monitoring. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 546 –552, July 2011.

[92] S. Paul, S. Nandi, and I. Singh. A dynamic balanced-energy sleep scheduling scheme in heterogeneous wireless sensor network. *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, pages 1–6, 2008.

[93] W. Qiu, M. Pham, and E. Skafidas. Routing and localization for extended lifetime in data collection wireless sensor networks. In *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pages 1373 –1378, Aug. 2008.

[94] D. Shuman and M. Liu. Optimal sleep scheduling for a wireless sensor network node. In *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, pages 1337–1341, 2006.

[95] C. Sibo, L. Zhiwei, and H. Hong. Wireless sensor network node localization algorithm based on adjacent node relationship. In *Communication Software*

and Networks (ICCSN), 2011 IEEE 3rd International Conference on, pages 215 – 218, May 2011.

[96] W. Stallings. *Operating Systems.* Prentice Hall, 2nd edition, 1995.

[97] Amulya R. Swain, R. C. Hansdah, and Vinod K. Chouhan. An energy aware routing protocol with sleep scheduling for wireless sensor networks. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 933–940, 2010.

[98] F. Tang, M. Guo, M. Li, Z. Wang, and Z. Cheng. Scalable and secure routing for large-scale sensor networks. *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, 2:300–305, 2008.

[99] F. Tang, M. Guo, M. Li, Y. Yang, D. Zhang, and Y. Wang. Wireless mesh sensor networks in pervasive environment: a reliable architecture and routing protocol. *Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on*, pages 72–72, 2007.

[100] S. Tian, X. Zhang, X. Wang, P. Sun, and H. Zhang. A selective anchor node localization algorithm for wireless sensor networks. In *Convergence Information Technology, 2007. International Conference on*, pages 358–362, 2007.

[101] TinyOS:. http://webs.cs.berkeley.edu/tos. Access on: June 2010.

[102] F. Tirkawi and S. Fischer. Adaptive tasks balancing in wireless sensor networks. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–6, 2008.

[103] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[104] Y. Tselishchev, L. Libman, and A. Boulis. Energy-efficient retransmission strategies under variable tdma scheduling in body area networks. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 374 – 381, Oct. 2011.

[105] I. Urteaga, N. Yu, N. Hubbell, and Q. Han. AWARE: Activity aware network clustering for wireless sensor networks. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 589 –596, Oct. 2011.

[106] Mei W. and M. Collier. TSEP: A localized algorithm for extending the lifetime of sensor networks. In *Wireless Advanced (WiAd), 2011*, pages 289 – 294, June 2011.

[107] Y. Wang and B. Ramamurthy. Layered clustering communication protocol for wireless sensor networks. *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 844–849, 2007.

[108] Y. Wang, D. Wang, W. Fu, and D. P. Agrawal. Hops-based sleep scheduling algorithm for enhancing lifetime of wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 709–714, 2006.

[109] Y. H. Wang, Y. L. Wu, and K. F. Huang. A power saving sleep scheduling based on transmission power control for wireless sensor networks. In *Ubi-Media Computing (U-Media), 2011 4th International Conference on*, pages 19 –24, July 2011.

[110] J. Woo-Sung, L. Keun-Woo, K. Young-Bae, and P. Sang-Joon. A hybrid approach for clustering-based data aggregation in wireless sensor networks. *Digi-*

tal Society, 2009. ICDS '09. Third International Conference on, pages 112–117, 2009.

[111] Y. H. Wu and W. M. Chen. Localization of wireless sensor networks using a moving beacon with a directional antenna. In *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, pages 237–242, 2009.

[112] Y. Xiao, H. Chen, K. Wu, B. Sun, Y. Zhang, X. Sun, and C. Liu. Coverage and detection of a randomized scheduling algorithm in wireless sensor networks. *Computers, IEEE Transactions on*, 59(4):507 – 521, April 2010.

[113] G. Xing, C. Lu, R. Pless, and Q. Huang. Impact of sensing coverage on greedy geographic routing algorithms. *Parallel and Distributed Systems, IEEE Transactions on*, 17(4):348 – 360, april 2006.

[114] J. Xu, M. Ma, and C. L. Law. AOA cooperative position localization. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5, 2008.

[115] J. Xu, G. Yang, Z. Y. Chen, L. Chen, and Z. Yang. Performance analysis of data aggregation algorithms in wireless sensor networks. In *Electrical and Control Engineering (ICECE), 2011 International Conference on*, pages 4619 –4622, Sept. 2011.

[116] X. Xu, Y. H. Hu, J. Bi, and W. Liu. Adaptive nodes scheduling approach for clustered sensor networks. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 34–39, 2009.

[117] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang. A delay-efficient algorithm for

data aggregation in multihop wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(1):163 – 175, Jan. 2011.

[118] M. Yu, S. J. Xiahou, and X. Y. Li. A survey of studying on task scheduling mechanism for tinyos. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–4, 2008.

[119] X. Yu, X. Xiaosong, and W. Wenyong. Priority-based low-power task scheduling for wireless sensor network. In *Autonomous Decentralized Systems, 2009. ISADS '09. International Symposium on*, pages 1–5, 2009.

[120] B. Zeng, Y. Dong, and D. Lu. Cooperation-based scheduling algorithm in wireless multimedia sensor networks. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 – 4, Sept. 2011.

[121] B. Zhang, F. Yu, and Z. Zhang. A high energy efficient localization algorithm for wireless sensor networks using directional antenna. In *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, pages 230–236, 2009.

[122] B. Zhang, F. Yu, and Z. Zhang. An improved localization algorithm for wireless sensor network using a mobile anchor node. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 2, pages 485–488, 2009.

[123] J. Zhang, F. Ren, T. He, and C. Lin. Data aggregation protocol based on dynamic routing in wireless sensor networks. *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, 1:501–505, 2009.

[124] L. Zhang and B. Deng. A new range-based localization algorithm for wireless sensor networks. In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, volume 1, pages 111 – 114, 2009.

[125] X. Zhang and B. Hu. An improved dv-hop algorithm using hop-count information and geometric constraint. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 – 4, Sept. 2011.

[126] W. Zhao and X. Tang. Scheduling data collection with dynamic traffic patterns in wireless sensor networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 286 – 290, April 2011.

[127] Y. Zhao, Q. Wang, W. Wang, D. Jiang, and Y. Liu. Research on the priority-based soft real-time task scheduling in tinyos. In *Information Technology and Computer Science, 2009. ITCS 2009. International Conference on*, volume 1, pages 562–565, 2009.

[128] Y. Zhao, J. Wu, F. Li, and S. Lu. VBS: Maximum lifetime sleep scheduling for wireless sensor networks using virtual backbones. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5, 2010.

[129] G. Zhou, T. He, S. Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys*, 2004.

[130] Y. Zhu, B. Zhang, F. Yu, and S. Ning. A RSSI based localization algorithm using a mobile anchor node for wireless sensor networks. In *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, volume 1, pages 123 – 126, april 2009.