ROUTING AND DATA DISSEMINATION IN WIRELESS SENSOR NETWORKS

THESIS SUBMITTED BY

Sanjeev Kumar Gupta
Reg. No. 2K 07-NITK-Ph.D.-1152-Co.

Submitted in partial fulfillment of the Requirements for the award of the degree of

DOCTORATE OF PHILOSOPHY

in

COMPUTER ENGINEERING

Under The Supervision of

Dr. Mayank Dave
Associate Professor
NIT, Kurukshetra

Department of Computer Engineering
National Institute of Technology
Kurukshetra-136119 India

August 2010
CANDIDATE’S DECLARATION

I hereby certify that the work which is being represented in the thesis entitled “ROUTING AND DATA DISSEMINATION IN WIRELESS SENSOR NETWORKS” in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Computer Engineering of National Institute of Technology, Kurukshetra is an authentic record of my own work carried out during a period from February 2007 to July 2010, under the supervision of Dr. Mayank Dave, Associate Professor, National Institute of Technology, Kurukshetra.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute/University.

(Sanjeev Kumar Gupta)
Reg. No. 2K 07-NITK-Ph.D.-1152-Co.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dated:

(Dr. Mayank Dave)
Associate Professor
NIT Kurukshetra
ABSTRACT

A wireless sensor network (WSN) extends our capability to explore, monitor and control the physical world. It is especially useful in catastrophic or emergency scenario where human participation may be too dangerous. The sensor networks have evolved over a period of time. The failures are inevitable in wireless sensor networks due to inhospitable environment and unattended deployment; therefore sensor nodes must operate potentially in large numbers. The latest generation of sensors encompasses self-organizing, flexible and scalable networks.

The energy is a critical factor in order to extend the lifetime of the network as nodes once deployed cannot be recharged. In WSN to reduce energy consumption of sensor nodes clustering is generally used. The clustering besides reducing energy consumption also helps in achieving efficient and scalable control. In an efficient clustering approach, radio communication distance should be minimized. In applications based on large scale WSNs that requires scalability to hundreds or thousands of nodes, usage of hierarchical clustering will be extremely useful.

In mission critical applications, data must be stored first, at least temporarily; within the network until either it is later collected by an observer or ceases to be useful. In case of crisis situations, automatic action in the target area is required. The action should also be performed in a timely manner otherwise it will be useless. Effective decision making especially in a crisis situation largely depends upon immediate access and interpretation of local information within the context of the overall environment at any particular point of time. Ensuring data availability is therefore very important especially in situations where there is a high risk of collapse of various components of WSN due to any inevitable disaster or enemy attack. In case the only storing and controlling data entity as used in most of the existing architectures is destroyed it will result in uselessness of the entire network. A new model based on an energy efficient real-time approach therefore has been proposed to overcome the problem of data availability in such situations.
The proposed architecture is expected to provide various features like timeliness guarantee, fault-tolerance and data integrity together with data-centric and distributed storage, besides enhanced energy efficiency. The proposed architecture is expected to help in faster and comprehensive decision making and automatic actions taken on the basis of stored data. In the architecture sensor node works either as a normal node or as a cluster head.

The proposed architecture is based on a cellular framework. The entire data of a cell is stored in a base station located at the centre of the cell and for data dissemination and actions, action and relay stations (ARSs) are used. Data centric storage mechanism is used for storing information. Data centric storage mechanism relies on naming the data instead of its location. Data sensed by various sensor nodes is also stored in a replicated manner to enhance its availability. The WSN based on proposed architecture consists of a single sink and many cells. The sink acts as a bridge between WSN and physical world. It supervises and synchronizes working of all other components of the proposed model.

Simulation studies were done to evaluate the performance of the proposed model on the basis of factors like average energy dissipation, number of messages received, deadline miss ratio and fault tolerance. Results obtained as a result of simulation were analyzed for network lifetime estimation, deadline miss ratio and fault tolerance.

In evaluation of the proposed model, centralized clustering algorithm has been used in case of small coverage area and results are compared with centralized clustering algorithm LEACH-C. In case of large coverage area, multi-level clustering has been used and obtained results have been compared with hierarchical algorithm HEED for network lifetime estimation. It has also been compared with real time protocol RAP for determining its effectiveness in achieving energy efficient real time communication. Results confirmed that the proposed architecture achieves better deadline miss ratio and enhances lifetime of the WSN. The proposed architecture is designed to operate under adverse conditions. It will continue to work unabatedly even if some critical components of the model are destroyed in the crisis situation.
DEDICATION

This thesis is dedicated to my father Dr. Rati Ram Gupta whose unwavering love, guidance, support and encouragement have enabled me to reach so far. He is like a light in cloud of uncertainty and difficulty. I wish I could become a little like him.
ACKNOWLEDGEMENTS

First of all I would like to express my sincere gratitude to my faculty advisor, Dr. Mayank Dave, Associate Professor, National Institute of Technology, Kurukshetra, for his earnest guidance and support throughout the course of this research. He has been a very brilliant advisor providing me the moral support, persistent encouragement and perpetual ideas. His depth of knowledge and enthusiasm for research has inspired me a lot. I really thank him heartedly for the precious hours he has spent with me discussing everything regarding this research to analyzing my research papers and critiquing my technical activities. Working with Dr. Mayank Dave has enriched my life professionally, intellectually and morally. His vision for the future has been an inspiration for me.

While at NIT, I had the privilege of interacting with bright and talented people. It was a great pleasure for me to work with all of them sincerely. I have received timely guidance and great cooperation from Dr. A.K. Singh, Dr. J.K. Chhabra, Dr. S.K. Jain and many other teachers. I also endure for the precious help rendered by Shri B.S. Kadian, Deputy Registrar of NIT Kurukshetra and all other officials/technocrats of the Department of Computer Engineering of NIT Kurukshetra.

I am indebted to my parents Dr. Rati Ram Gupta and Mrs. Nirmal Gupta for everything that they have given me too much. They have taught me the value of knowledge, the joy of love and the importance of sincere family activities. They have stood by me in every thick and thin. I have done anything so far under their constant support and hearted affection. They are a great inspiration to me in all that they could provide to me throughout my life. I also want to thank my brothers Rajeev Gupta, Anil Gupta and Parveen Gupta for their timely help and moral encouragement.

Finally, I would like to extend my hearted thanks to my dear wife Anita Gupta and my lovely sons Anmol and Gaurav, whose unwavering love, understanding and sacrifices during the period of my research. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness. I know that there is nothing, I cannot accomplish, with them by my side. I desire to express my gratitude and sincerity to many other persons which could not be mentioned here.

Sanjeev Kumar Gupta
# TABLE OF CONTENTS

CANDIDATE’S DECLARATION................................................................................................. i  
ABSTRACT ........................................................................................................................ ii  
DEDICATION....................................................................................................................... iv  
ACKNOWLEDGEMENTS ..................................................................................................... v  
ABBREVIATIONS ............................................................................................................. xii  
LIST OF TABLES ................................................................................................................ xiii  
LIST OF FIGURES ............................................................................................................. xiv  

CHAPTER 1 ......................................................................................................................... 1  
INTRODUCTION TO WIRELESS SENSOR NETWORKS ........................................... 1  
1.1 INTRODUCTION ...................................................................................................... 1  
1.2 WIRELESS SENSOR NETWORKS CHARACTERISTICS ...................................... 3  
1.3 ADVANTAGES OF WIRELESS SENSOR NETWORKS ........................................ 4  
1.4 CHALLENGES IN WIRELESS SENSOR NETWORKS .......................................... 5  
1.5 GENERATIONS OF SENSOR NETWORKS .......................................................... 8  
1.6 STORAGE MANAGEMENT ...................................................................................... 8  
  1.6.1 Storage management components................................................................. 9  
  1.6.2 Indexing and data retrieval ............................................................................. 10  
  1.6.3 Advantages of storage management ............................................................ 10  
1.7 APPLICATION AREAS OF WIRELESS SENSOR NETWORKS ...................... 11  
1.8 RESEARCH MOTIVATION AND CHALLENGES ............................................. 14  
1.9 THESIS CONTRIBUTION ..................................................................................... 16  
1.10 THESIS OUTLINE ............................................................................................... 19  

CHAPTER 2 ......................................................................................................................... 20  
REVIEW OF LITERATURE AND GENERAL CONSIDERATIONS ......................... 20  
2.1 INTRODUCTION ..................................................................................................... 20  
2.2 WIRELESS SENSOR AND ACTOR NETWORKS (WSAN) ................................... 20
2.3 SURVEY OF EXISTING WORK ON REAL TIME PROTOCOL IN WSN....... 22
   2.3.1 RAP .............................................................................................................. 22
   2.3.2 SPEED ......................................................................................................... 24
   2.3.3 MM-SPEED ................................................................................................. 26
   2.3.4 RPAR ........................................................................................................... 27
   2.3.5 DGR ............................................................................................................. 29
   2.3.6 ACM ............................................................................................................ 29
   2.3.7 Other Protocols ............................................................................................ 30
2.4 SURVEY OF EXISTING WORK ON CLUSTERING ......................................... 34
   2.4.1 Random Competition Based Clustering (RCC) ........................................... 35
   2.4.2 Banerjee and Khuller ................................................................................... 35
   2.4.3 GS\(^3\) ........................................................................................................... 36
   2.4.4 Energy Efficient Hierarchical Clustering (EEHC) ....................................... 37
   2.4.5 Low Energy Adaptive Clustering Hierarchy (LEACH) .............................. 38
   2.4.6 LEACH-C .................................................................................................... 39
   2.4.7 Hybrid Energy-Efficient Distributed Clustering (HEED) ......................... 39
2.5 CONCLUSION ....................................................................................................... 42

CHAPTER 3 .................................................................................................................... 43
DISTRIBUTED CELLULAR ARCHITECTURE FOR DATA PLACEMENT IN
WIRELESS SENSOR NETWORKS ............................................................................. 43
3.1 INTRODUCTION ................................................................................................. 43
3.2 ARCHITECTURAL FRAMEWORK ...................................................................... 43
   3.2.1 Cluster management .................................................................................... 44
   3.2.2 Sensor-ARS coordination ......................................................................... 47
   3.2.3 Node placement .......................................................................................... 48
   3.2.4 Localization methods ................................................................................ 48
   3.2.5 Addressing scheme .................................................................................... 50
   3.2.6 Data routing ............................................................................................... 50
   3.2.7 Data storage ............................................................................................... 51
   3.2.8 Distributed storage and replication ............................................................ 52
4.6.4 Performance of Sensor Node ................................................................. 73
4.7 MODEL OF CLUSTER ........................................................................... 73
  4.7.1 Re-clustering Process ................................................................. 74
  4.7.2 Data Objects of Cluster Class .................................................. 74
4.8 MODEL OF ARS ................................................................................... 77
  4.8.1 Working of ARS ................................................................. 78
  4.8.2 Data Objects of ARS Class ..................................................... 78
4.9 MODEL OF ACTION UNIT CLASS ...................................................... 80
  4.9.1 Data Objects of Action Unit Class ............................................. 80
4.10 MODEL OF BASE STATION ............................................................... 80
  4.10.1 Working of Base Station ......................................................... 81
  4.10.2 Data Objects of Base Station Class ........................................... 83
4.11 MODEL OF CELL ............................................................................... 84
  4.11.1 Data Objects of Cell class ....................................................... 84
4.12 MODEL OF SINK ............................................................................... 84
  4.12.1 Working of Sink ................................................................. 84
  4.12.2 Data Objects of Sink class ..................................................... 85
4.13 MODEL OF RTWSN .......................................................................... 85
  4.13.1 Data Objects of RTWSN class ................................................ 85
4.14 FORMAL VERIFICATION USING SPIN .............................................. 86
4.15 CONCLUSION .................................................................................... 86

CHAPTER 5 .................................................................................................. 87
PARAMETERIC ESTIMATION FOR MODEL SIMULATION ..................... 87
  5.1 INTRODUCTION .............................................................................. 87
    5.1.1 Formulation Notation .............................................................. 87
    5.1.2 Decision Variables ................................................................. 88
    5.1.3 Amplifier Energy ................................................................. 88
    5.1.4 Clustering ........................................................................... 89
5.2 TIME MEASUREMENT ........................................................................ 89
    5.2.1 Time Synchronization ............................................................ 91
6.6 SIMULATION RESULTS FOR NETWORK LIFETIME ESTIMATION FOR LARGE COVERAGE AREA OF A CELL ................................................................. 130
6.7 SIMULATION RESULTS FOR DEADLINE MISS RATIO FOR LARGE COVERAGE AREA OF A CELL ................................................................. 133
  6.7.1 Changing CH after every round and choosing remaining battery level as major factor for electing CH ................................................................. 134
  6.7.2 Changing CH after every round and choosing distance from controlling authority as major eligibility factor for electing CH ........................................ 137
  6.7.3 Changing CH only after its battery level falls below a threshold value and choosing distance from destination as major factor for electing CH .......... 141
6.8 SIMULATION RESULTS FOR FAULT TOLERANCE .................................... 145
6.9 CONCLUSION ................................................................................................. 149

CHAPTER 7 ................................................................................................................. 151
CONCLUSION AND FUTURE DIRECTIONS ........................................................ 151
  7.1 CONCLUSION ................................................................................................. 151
  7.2 FUTURE DIRECTIONS ................................................................................ 153

ANNEXURE A .............................................................................................................. 155
DATA STRUCTURES USED IN PROPOSED MODEL .............................................. 155

REFERENCES ......................................................................................................... 160

PUBLICATIONS ........................................................................................................ 175
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>Assured Corridor Mechanism</td>
</tr>
<tr>
<td>ARS</td>
<td>Action and Relay Station</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster Heads</td>
</tr>
<tr>
<td>DGR</td>
<td>Directional Geographical Routing protocol</td>
</tr>
<tr>
<td>EEHC</td>
<td>Energy Efficient Hierarchical Clustering</td>
</tr>
<tr>
<td>FDT</td>
<td>Formal Description Techniques</td>
</tr>
<tr>
<td>GHT</td>
<td>Geographic Hash Table</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HEED</td>
<td>Hybrid Energy-Efficient Distributed Clustering</td>
</tr>
<tr>
<td>JSPIN</td>
<td>Java GUI for SPIN</td>
</tr>
<tr>
<td>LEACH</td>
<td>Low Energy Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>LEACH-C</td>
<td>LEACH Centralized</td>
</tr>
<tr>
<td>LTL</td>
<td>Linear Temporal Logic</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electro-Mechanical System</td>
</tr>
<tr>
<td>PROMELA</td>
<td>Process Meta Language</td>
</tr>
<tr>
<td>QOS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAP</td>
<td>A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks</td>
</tr>
<tr>
<td>RCC</td>
<td>Random Competition Based Clustering</td>
</tr>
<tr>
<td>RPAR</td>
<td>Real-time Power-Aware Routing protocol</td>
</tr>
<tr>
<td>SPEED</td>
<td>A Stateless Protocol for Real-Time Communication in Sensor Networks</td>
</tr>
<tr>
<td>SPIN</td>
<td>Simple Model Interpreter</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiplexing</td>
</tr>
<tr>
<td>UFLP</td>
<td>Uncapacitated Facility Location Problem</td>
</tr>
<tr>
<td>VIP</td>
<td>Visual Interface to PROMELA</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WSAN</td>
<td>Wireless Sensor and Actor Network</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1.1 Comparisons between various wireless networks ............................................. 3
Table 2.1 Classification of RT routing protocols in sensor networks ....................... 33
Table 2.2 Classification of clustering protocols in sensor networks ......................... 41
Table 6.1 Simulation Parameters for small coverage area of the cell ...................... 105
Table 6.2 Centralized Clustering Algorithm .............................................................. 106
Table 6.3 Simulation Parameters for large coverage area of the cell ...................... 128
Table 6.4 Hierarchical Clustering ........................................................................... 129
LIST OF FIGURES

Fig. 1.1 Applications of Wireless Sensor Networks .......................................................... 12
Fig. 2.1 An illustration of an integrated architecture of WSANs [22] ......................... 21
Fig. 2.2 RAP communication architecture [26] ............................................................ 23
Fig. 2.3 Routing components of SPEED [29] ............................................................... 25
Fig. 2.4 Queuing model on a particular sensor node [42] ........................................... 30
Fig. 2.5 An Example of a Three Layer Cluster Hierarchy [56] .................................. 36
Fig. 3.1 Clustering for Small Coverage Area of Cell ................................................. 44
Fig. 3.2 Clustering for Large Coverage Area of Cell ................................................ 45
Fig. 3.3 Centralized Clustering Process ..................................................................... 46
Fig. 3.4 Finding point’s position knowing distance to three known points. [64] ....... 49
Fig. 3.5 Cellular Network Framework ....................................................................... 51
Fig. 3.6 Proposed System Architecture ..................................................................... 55
Fig. 3.7 State Transition at each sensor node ............................................................. 57
Fig. 4.1 Usual Way of Software Development [93] ................................................... 62
Fig. 4.2 Software Development using Formal Description Techniques [93] .......... 63
Fig. 4.3 Skeleton of a PROMELA Program ............................................................... 65
Fig. 4.4 Example for Variable and Channel Declarations ......................................... 66
Fig. 4.5 Non-Deterministic Possibilities in a Finite State Machine ......................... 67
Fig. 4.6 Working of Cluster Head .............................................................................. 71
Fig. 4.7 Working of Normal Sensor Node ................................................................. 73
Fig. 4.8 Re-Clustering ............................................................................................... 75
Fig. 4.9 Structure of Cluster Class ............................................................................ 76
Fig. 4.10 Finite State Machine of ARS ................................................................. 77
Fig. 4.11 Working of ARS ...................................................................................... 79
Fig. 4.12 Structure of Action Unit ......................................................................... 80
Fig. 4.13 Finite State Machine for Base Station ...................................................... 81
Fig. 4.14 Working of Base Station .......................................................................... 82
Fig. 4.15 Structure of Cell ....................................................................................... 83
Fig. 4.16 Working of Sink ......................................................................................... 84
Fig. 4.17 Structure of RTWSN

Fig. 5.1 Relationship between the Measured time and Actual time [106]

Fig. 5.2 Two-way message handshake. [107]

Fig. 5.3 (a) Probabilistic sensing model; (b) communication model. [110]

Fig. 5.4 Lifetime of WSN with respect to c [63]

Fig. 5.5 Virtual Queues at Receiver [114]

Fig. 6.1 Remaining Average Battery Level per Node over various Rounds

Fig. 6.2 Number of Live Nodes over various Rounds

Fig. 6.3 Number of Messages Received over various Rounds

Fig. 6.4 Number of Messages received per ARS over various Rounds

Fig. 6.5 Number of Messages Received over various Rounds

Fig. 6.6 Number of Messages received per ARS over various Rounds

Fig. 6.7 Number of Live Nodes over various Rounds

Fig. 6.8 Average Battery Consumed per Node over various Rounds

Fig. 6.9 Average Battery Consumed per Node over various Rounds

Fig. 6.10 Number of Messages Received over various Rounds

Fig. 6.11 Number of Live Nodes over various Rounds

Fig. 6.12 Number of Messages received per ARS over various Rounds

Fig. 6.13 Average Battery Consumed per Node over various Rounds

Fig. 6.14 Number of Messages received per ARS over various Rounds

Fig. 6.15 Number of Messages per Cell over various Rounds

Fig. 6.16 Number of Messages Received over various Rounds

Fig. 6.17 Number of Live Nodes over various Rounds

Fig. 6.18 Number of Messages received per ARS over various Rounds

Fig. 6.19 Number of Messages per Cell over various Rounds

Fig. 6.20 Number of Messages Received over various Rounds

Fig. 6.21 Remaining Average Battery Level per Node over various Rounds

Fig. 6.22 Number of Live Nodes over various Rounds

Fig. 6.23 Number of Messages per Cell over various Rounds

Fig. 6.24 Number of Messages received per ARS over various Rounds

Fig. 6.25 Number of Messages Received over various Rounds
Fig. 6.26 Remaining Average Battery Level per Node over various Rounds ........ 125
Fig. 6.27 Number of Live Nodes over various Rounds .............................................. 125
Fig. 6.28 Average Battery Consumed per Node over various Rounds ..................... 131
Fig. 6.29 Number of Messages Received over various Rounds ................................. 131
Fig. 6.30 Average Battery Consumption of Nodes having BS within Sensing Radius ......................................................................................................................................... 132
Fig. 6.31 Percentage of Messages which have achieved Deadline.......................... 134
Fig. 6.32 Average Number of Hops per Message ....................................................... 135
Fig. 6.33 Average Battery Consumed per Node over various Rounds ..................... 135
Fig. 6.34 Remaining Battery Level of Nodes having BS within Sensing Radius .... 136
Fig. 6.35 Number of Messages over various Rounds ................................................. 136
Fig. 6.36 Percentage of Messages which have achieved Deadline.......................... 138
Fig. 6.37 Average Number of Hops per Message ....................................................... 138
Fig. 6.38 Number of Messages over various Rounds ................................................. 139
Fig. 6.39 Average Battery Consumed per Node over various Rounds ..................... 139
Fig. 6.40 Remaining Battery Level of Nodes having BS within Sensing Radius .... 140
Fig. 6.41 Average Battery Consumed per Node over various Rounds ..................... 142
Fig. 6.42 Percentage of Messages which have achieved Deadline.......................... 142
Fig. 6.43 Average Number of Hops per Message ....................................................... 143
Fig. 6.44 Number of Messages over various Rounds ................................................. 143
Fig. 6.45 Remaining Battery Level of Nodes having BS within Sensing Radius .... 144
Fig. 6.46 Messages received per Base-Station ............................................................ 146
Fig. 6.47 Average Remaining Battery of nodes ......................................................... 147
Fig. 6.48 Average Battery Consumption per Node .................................................... 147
Fig. 6.49 Avg. Battery consumption of nodes within sensing radius of destination 148
CHAPTER 1
INTRODUCTION TO WIRELESS SENSOR NETWORKS

1.1 INTRODUCTION

Wireless Sensor Networks (WSNs) [1]-[2] have emerged as research areas with an overwhelming effect on practical application developments. They permit fine grain observation of the ambient environment at an economical cost much lower than currently possible. In hostile environments where human participation may be too dangerous sensor networks may provide a robust service. Sensor networks are designed to transmit data from an array of sensor nodes to a data repository on a server. The advances in the integration of micro-electro-mechanical system (MEMS), microprocessor and wireless communication technology have enabled the deployment of large-scale wireless sensor networks. WSN has potential to design many new applications for handling emergency, military and disaster relief operations that requires real time information for efficient coordination and planning.

Sensors are devices that produce a measurable response to a change in a physical condition like temperature, humidity, pressure etc. WSNs [3] may consist of many different types of sensors such as seismic, magnetic, thermal, visual, infrared, acoustic and radar, capable to monitor a wide variety of ambient conditions. Though each individual sensor may have severe resource constraint in terms of energy, memory, communication and computation capabilities; large number of them may collectively monitor the physical world, disseminate information upon critical environmental events and process the information on the fly [4]-[7].

The issues of network lifetime and data availability are extremely important in WSN due to their deployment in hostile environment. The system should provide fault tolerant energy efficient real-time communication as well as automatic and effective action in crisis situations. A typical sensor network [8] operates in five phases which are planning phase, deployment phase, post-deployment phase, operation phase and post-operation phase.
a. In planning phase, a site survey is conducted to evaluate deployment environment and its conditions to select a suitable deployment mechanism.
b. In deployment phase, sensors are randomly deployed over a target region.
c. In post-deployment phase, the sensor networks operators need to identify or estimate the location of sensors to access coverage.
d. The operation phase involves the normal operation of monitoring tasks where sensors observe the environment and generate data.
e. The post-operation phase involves shutting down and preserving the sensors for future operations or destroying the sensor network.

The sensor nodes consist of sensing, data processing and communicating components. They can be used for continuous sensing, event detection as well as identification, location sensing and control of actuators. The nodes are deployed either inside the phenomenon or very close to it and can operate unattended. They can use their processing abilities to locally carry out simple computations and transmit only required and partially processed data. They may be organized into clusters or collaborate together to complete a task that is issued by the users. In addition, positions of these nodes do not need to be predefined. These allow their random deployment in inaccessible terrains or disaster relief operations.

The WSN provides an intelligent platform to gather and analyze data without human intervention. As a result, WSNs have a wide range of applications such as military applications [7], to detect and track hostile objects in a battle field or in environmental research applications [9], to monitor a disaster as seismic tremor, a tornado or a flood or for industrial applications [10], to guide and diagnose robots or machines in a factory or for educational applications [11], to monitor developmental childhood or to create a problem-solving environment [12].

The wireless sensor nodes are generally battery driven and due to their deployment in harsh or hostile environment their battery is usually un-chargeable and un-replaceable. Moreover, since their sizes are too small to accommodate a large battery, they are constrained to operate using an extremely limited energy budget. The total stored energy in a
smart dust mote, for instance is only 1J [13]. Since this small amount of energy is the only power supply to a sensor node, it plays a vital role in determining lifetime of the sensor networks. All the research works therefore have a common concern of minimizing energy consumption and it is a significant issue at all layers of the WSN. Other key issues are scalability to large number of nodes, design of data handling techniques, localization techniques, real time communication, data availability, fault tolerance etc.

1.2 WIRELESS SENSOR NETWORKS CHARACTERISTICS

A WSN is different from other popular wireless networks like cellular network, wireless local area network (WLAN) and Bluetooth in many ways. Compared to other wireless networks, a WSN has much more nodes in a network, distance between the neighbouring nodes is much shorter and application data rate is much lower also. Due to these characteristics, power consumption in a sensor network should be minimized. Table 1.1 compares the characteristics of these networks as under:

<table>
<thead>
<tr>
<th>Types</th>
<th>No. of Nodes</th>
<th>Range</th>
<th>Data Rate</th>
<th>Mobility</th>
<th>Power</th>
<th>Redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cellular</td>
<td>Large</td>
<td>Long</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>WLAN</td>
<td>Small</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>Small</td>
<td>Short</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>WSN</td>
<td>Large</td>
<td>Very Short</td>
<td>Low</td>
<td>Low</td>
<td>Very Low</td>
<td>High</td>
</tr>
</tbody>
</table>

To keep the cost of the entire sensor network down, cost of each sensor needs to be reduced. It is also important to use tiny sensor nodes. A smaller size makes it easier for a sensor to be embedded in the environment it is in. WSNs may also have a lot of redundant data since multiple sensors can sense similar information. The sensed data therefore need to be aggregated to decrease the number of transmissions in the network, reducing bandwidth usage and eliminating unnecessary energy consumption in both transmission and reception.
1.3 ADVANTAGES OF WIRELESS SENSOR NETWORKS

The WSNs has revolutionized the world around us. They are becoming integral part of our lives, more so than the present-day computers because of their numerous advantages as mentioned below:-

i. **Ease of deployment**

A sensor network contains hundreds or even thousands of nodes and can be deployed in remote or dangerous environments. Since these nodes are small and economical, throwing of hundreds or thousands of micro-sensors from a plane flying over a remote or dangerous area allows extracting information in ways that could not have been possible otherwise.

ii. **Extended range of sensing**

Single macro-sensor nodes can only extract data about events in a limited physical range. In contrast, a micro-sensor network uses large numbers of nodes enabling them to cover a wide area.

iii. **Improved lifetime**

The nodes located close to each other will have correlated data therefore they can be grouped together. Only one of the nodes in a round robin fashion from the group therefore needs to be in active state at any instance of time keeping other nodes in sleep state. It will enhance the network life time.

iv. **Fault tolerance**

In WSN several sensor nodes are close to each other and have correlated data, it makes these systems much more fault tolerant than single macro-sensor system. The macro-sensor system cannot function if macro-sensor node fails, whereas in case of micro-senor network even if smaller number of micro-sensor nodes fails, the system may still produce acceptable qualitative information.

v. **Improved accuracy**

While an individual micro-sensor’s data might be less accurate than a macro-sensor’s data. The data from nodes located close to each other can be combined since they are gathering information about the same event. It will result in better accuracy of the sensed data and reduced uncorrelated noise.
vi. **Lower cost**

Even though, to replace each macro-sensor node several micro-sensor nodes are required they will still be collectively much cheaper than their macro-sensor counterpart due to their reduced size, simple as well as cheap circuitry and lesser accuracy constraints. As a result protocols that enable micro-sensor networks to provide necessary support in sensing applications are becoming more popular.

vii. **Actuation**

Actuation can dramatically extend the capabilities of a sensor network in two ways. First, it can enhance the sensing task, by pointing cameras, aiming antennae or repositioning sensors. Secondly, it can affect the environment – by opening valves, emitting sounds or strengthening beams [14].

viii. **Collaborative objective**

Perhaps the most important aspect of sensor networks that differentiates them from other wireless networks is their objective. Typically, objective of a sensor network is monitoring a specific signal of interest and informing a central base station or a sink about activities in the region being sensed. Since a sensor network is deployed for achieving a certain system-wide goal, nodes collaborate instead of competing with each other.

1.4 **CHALLENGES IN WIRELESS SENSOR NETWORKS**

In order to design good applications for wireless micro-sensor networks, it is essential to understand factors important to the sensor network applications. Although WSNs share some commonalities with existing wireless ad-hoc networks they pose a number of technical challenges different from traditional wireless ad-hoc networks [4]-[6]. The protocols and algorithms that have been proposed for traditional wireless ad-hoc networks are therefore not well suited for the application requirements of the sensor networks. To illustrate this point, differences between sensor networks and traditional networks are outlined below:

i. **Energy**

The sensor nodes are generally inaccessible after deployment and normally they have a finite source of energy that must be optimally used for processing and communication to extend their lifetime. It is a well known fact that communication requires significant energy.
In order to make optimal use of energy, therefore communication should be minimized as much as possible.

**ii. Redundancy**

Due to the frequent node failures and inaccessibility of failed nodes, WSNs are required to have high redundancy of nodes so that the failure of few nodes can be negligible.

**iii. System lifetime**

The WSNs should function as long as possible. Their system lifetime can be measured by using generic parameters such as time until the nodes die or by using application specific parameters like time until the sensor network is no longer providing acceptable quality results.

**iv. Scalability**

In WSNs, each sensor node obtains a specific view of the environment. A given sensor's view of the environment is limited both in range and accuracy; it can only cover a limited physical area of the environment. The WSNs therefore, deploys sensor nodes that have a short transmission distance in large numbers to monitor the entire area.

**v. Adaptability**

The WSN system should be adaptable to changes such as addition of more nodes, failure of nodes, environmental conditions and thus unlike traditional networks, where the focus is on maximizing channel throughput or minimizing node deployment, the major consideration in a sensor network is to extend the network lifetime besides system robustness.

**vi. Application awareness**

A WSN is not a general purpose network. In order to deploy it for specific application, the WSN protocols should consider application-specific trade-offs in terms of complexity, resource usage and communication patterns to improve network efficiency.

**vii. Lack of global identification**

Due to large number of sensor nodes in a sensor network the global identification (GID) is generally not possible. Although in some cases, the Global Positioning System (GPS) provides positioning information to sensor nodes but it requires line of sight to several satellites, which is generally not available inside of buildings, beneath dense foliage, underwater, when jammed by an enemy or during MARS exploration etc.
viii. Storage, search and retrieval

The sensor network can produce a large volume of raw data such as continuous time-series of observations over all points in space covered by the network. Since the data source is continuous traditional databases are not suitable for WSNs.

ix. Data centric processing

The naming schemes in WSNs are often data-oriented for example an environmental monitoring system may requests temperature readings through a query like “collect temperature readings in the region bounded by the rectangle (x1,y1,x2,y2)”, instead of a query “collect temperature readings from a set of nodes having addresses x, y and z.”

x. Production cost

The cost of a single node is very important to justify overall cost of the network; since the sensor networks consist of a large number of sensor nodes therefore cost of each sensor node has to be kept low.

xi. Node deployment

Node deployment is application dependent and affects performance of the protocol. The deployment is either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data is routed through pre-determined paths. However, in self organizing systems, the sensor nodes are scattered randomly creating an infrastructure in an ad-hoc manner.

xii. In-network processing

In general transport protocols used in wired and wireless networks have assumed end-to-end approach guaranteeing that data from the senders have not been modified by intermediate nodes until it reaches a receiver. However, in WSNs data can be modified or aggregated by intermediate nodes in order to remove redundancy of information. The previous solutions did not accommodate concept of in-network processing, called data aggregation or diffusion in WSNs.

xiii. Latency

Latency refers to delay from when a sender sends a packet until the packet is successfully received by the receiver. The sensor data has a temporal time interval in which it is valid, since the nature of the environment changes constantly, it is therefore important to receive the data in a timely manner.
xiv. Fault tolerance

Sensor nodes are fragile and they may fail due to depletion of batteries or destruction by an external event. Realizing a fault-tolerant operation is critical, for successful working of the WSN, since faulty components in a network lead to reduced throughput, thereby decreasing efficiency and performance of the network.

1.5. GENERATIONS OF SENSOR NETWORKS

In a similar fashion to the evolution of other technologies, it is possible to describe the evolution of sensor networks in terms of generations.

- First Generation Sensor Networks (1GSN)
  A sensor network consists of individual sensor devices. Deployment is via manual emplacement. The network is fully preconfigured. Access to information is via manual retrieval of the device itself or long-range point-to-point communication links.

- Second Generation Sensor Networks (2GSN)
  Sensors work in collaboration to cover an area. The network typically consists of a small number of sensors communicating with a control node equipped with a reach-back link. They are typically manually deployed, relying heavily on pre-configuration.

- Third Generation Sensor Networks (3GSN)
  The third generation of sensors encompasses self-organizing, flexible and scalable networks. Sensors communicate with one another for two purposes: communications services (e.g. automatic relaying of messages to a network gateway) and in-network processing (data aggregation and data fusion).

1.6 STORAGE MANAGEMENT

Storage management is an area of sensor network research that is attracting attention. In such class of sensor networks, the data must be stored, at least temporarily; within the network until either it is later collected by an observer or ceases to be useful. For example, consider a sensor network that is deployed in a military scenario collecting information about
nearby activity. The data has to be dynamically queried by soldiers to attain the mission goals or avoiding sources of danger and help the commanders to assess progress of the mission. The queried data are real-time as well as long-term about enemy activity (for example, to answer a question: Where the supply lines are located?). The data must be stored to enable queries that span temporally long periods, such as days or even months.

One can envision similar applications with sensor networks deployed in other contexts that answer questions about the environment using recent or historical data. In such networks collected data are later accessed by dynamically generated queries. With the knowledge of relevant application and system characteristics, a set of goals for the sensor network storage management can be determined:

a. Minimizing storage size to maximize coverage/data retention
b. Minimizing energy
c. Supporting efficient query execution on the stored data (note that in the reach-back method where all the data must be sent to the observer, query execution is simply transfer of the data to the observer)
d. Providing efficient data management under constrained storage.

Efficiency of query execution can be measured in terms of retrieval time, communication overhead and energy consumption required in sending requested data to the observer. The storage management can influence the efficiency of query execution by effective data placement and indexing. Several approaches to storage management have been proposed to meet these requirements, with most approaches involving a tradeoff among these different goals.

1.6.1 Storage management components

The storage management can be split into three major components:

- System support for storage management.
- Collaborative storage.
- Indexing and retrieval.
Specifically, each sensor has a local view of the phenomenon. Each sensor sends sensed data to the cluster head (CH) for collaborative storage. Each CH sends it further to resource rich destination for higher degree of collaboration. Further as a result of coordination, a significant reduction in the data to be stored is achieved.

1.6.2 Indexing and data retrieval

Indexing and retrieval are more important issues in military type applications, where data can be queried dynamically for example; a commander may be interested in enemy tank movements. Such networks are inherently data-centric; observers often name data in terms of attributes or content that may not be topologically relevant. This characteristic of such sensor networks is similar to many peer-to-peer (P2P) environments [15], which are also often data-centric. However, existing solutions of data indexing and retrieval in P2P networks are not suitable for sensor networks due to excessive communication required.

The Geographic Hash Table (GHT) [16] is one of the approaches which can be applied to sensor networks. The GHT is a structured approach to sensor network storage that makes it possible to index data on the basis of content without requiring query flooding. GHT also provides load balancing of storage usage (assuming fairly uniform sensor deployment). GHT implements a distributed hash table by hashing a key $k$ into geographic coordinates.

1.6.3 Advantages of storage management

Accessing and processing data produced in a wireless sensor network using a database-like approach [17]-[19] has several advantages. Sensors can be deployed in physical environment and applications that manipulate their data; can be created, refined and modified afterwards without any physical intervention on the sensors themselves. The data management activity performed in the network can be remotely controlled by interactively issuing queries, expressed in a high level language, which specify the data, those are of interest for a specific task and how these should be manipulated.
1.7 APPLICATION AREAS OF WIRELESS SENSOR NETWORKS

WSNs have opened the eye of new generation scientists to observe never before phenomenon, paving the way for designing of numerous applications. These applications of WSNs can be classified into three categories (as shown in Figure 1.1):

- Monitoring space
- Monitoring things
- Monitoring the interactions of things with each other and the encompassing space

Space monitoring includes environmental and habitat monitoring, precision agriculture, indoor climate control, surveillance, treaty verification and intelligent alarms. Whereas monitoring things includes structural monitoring, eco-physiology, condition-based equipment maintenance, medical diagnostics and urban terrain mapping. Further, the most dramatic applications of WSN involve monitoring complex interactions, including wildlife habitats, disaster management, emergency response, ubiquitous computing environments, asset tracking, manufacturing process flow and healthcare. The details of some of the major applications are briefly described below:-

a. Habitat monitoring

Researchers in the life sciences are becoming increasingly concerned about potential impacts of the human presence in monitoring plants and animals in field conditions for example the seabird colonies are notorious for their sensitivity to human disturbance. The WSNs therefore can be used to gather information on the habitat of a plant/animal without disturbing them. The gathered data can be analyzed later on to learn optimal environmental conditions favorable for the flora/fauna’s growth.

b. Military

The use of WSN can provide real time information of the enemy activities to commando teams thus making coordination and planning more effective. The sensing, monitoring and decision-making should be integrated seamlessly, for designing effective military applications. The accurate and timely gathering of visual surveillance and intelligence data can play a central role in attaining objectives as well as minimizing loss of human lives.
### Application Areas of Wireless Sensor Networks

<table>
<thead>
<tr>
<th>Application</th>
<th>Areas of Wireless Sensor Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental Observation</td>
<td></td>
</tr>
<tr>
<td>Precision Agriculture</td>
<td></td>
</tr>
<tr>
<td>Disaster Management</td>
<td></td>
</tr>
<tr>
<td>Healthcare Management</td>
<td></td>
</tr>
<tr>
<td>Military Surveillance</td>
<td></td>
</tr>
<tr>
<td>Habitat Monitoring</td>
<td></td>
</tr>
<tr>
<td>Home Networks</td>
<td></td>
</tr>
<tr>
<td>Pollution Control</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1.1 Applications of Wireless Sensor Networks
c. **Home Automation**

Networking various home appliances, such as vacuum cleaners, micro-wave ovens, and refrigerators, with wireless medium, has been dreamt of for many years. Embedded sensors inside such appliances can interact with each other, and with the external network via the internet or satellites. They allow users to manage home devices locally and remotely more easily.

d. **Precision Agriculture**

The WSNs monitors environmental conditions in which farming is done to make it more profitable and sustainable. The WSNs are proving useful for controlling in economical way climate, irrigation and nutrient supply to produce best crop condition, increase in production efficiency while decreasing cost. They are also helping in strategic planning and counter measures to increase yield of the crop.

e. **Healthcare**

Sensors are used in biomedical applications for healthcare. Sensors are implanted in the human body for monitoring medical problems such as cancer and help patients to maintain their health.

f. **Building monitoring**

Sensors can be used in buildings for detection of fire and smoke. In case of fire a network of sensors deployed in a huge building can track the source and direction in which fire is expanding. In addition, sensors can be used to monitor vibration that could damage the structure of a building.

g. **Environmental observation**

WSNs can be used to monitor environment such as forest fire detection, flood detection, air pollution detection, rainfall observation in agriculture etc. Sensor nodes can be used for detection of toxic waste, illegally dumped into the lake by a factory located nearby and relaying the exact origin of a pollutant to a centralized authority, which then can take appropriate measures, to limit spread of the pollution.
Without the WSN, it would be difficult to get the data without the nearby factory’s knowledge, in which case the factory would prevent the data gathering process.

**h. Disaster Management**

The reliable early warning system based on WSN can be deployed in areas with high risk of disasters. The use of WSN promise to provide real time information of the disaster area to rescue teams making coordination and planning more effective. Location information of victims, rescuers and objects in the disaster is vital for the rescue operations.

It has been known that, for an operationally effective disaster management: sensing, monitoring and decision-making should be integrated seamlessly. Timely and updated disaster information is extremely important for efficient response and effective actions, it will help disaster managers [20] make better decisions and take actions in time.

**1.8 RESEARCH MOTIVATION AND CHALLENGES**

The rising need of new information processing paradigm such as environmental monitoring and surveillance tasks have led to large scale active research in the fields of highly distributed sensor networks. This study is especially useful in catastrophic or emergency scenario where human participation may be dangerous. The failures are inevitable in WSNs due to inhospitable environment and unattended deployment; therefore sensor nodes must operate potentially in large numbers. The national security and disaster management theme provided the inner urge for this research in routing and data dissemination in WSNs.

There is always a need of an efficient architecture tailored towards enhancing network lifetime. The energy is a critical factor in order to extend lifetime of the network, since nodes once deployed cannot be recharged. The lifetime of a network is nothing but the time until a certain number (or percentage) of the sensor nodes run out of energy. The clustering is the standard approach [21] used for enhancing network lifetime. It results in significant reduction of energy consumption of member nodes of the cluster and also achieves efficient as well as scalable control by minimizing radio communication distances.
The next research challenge is the need to provide features like effective and automatic action on the sensed data besides reducing energy consumption in real world mission critical WSN based applications. The collected data is useless if it cannot be acted upon automatically and comprehensively in the critical area in a timely manner. The Wireless sensor and actuator networks (WSAN) [22] are becoming the most addressed research fields in the area of information and communication technologies these days. In a crisis situation effective decision making also depends upon immediate access and interpretation of the local information within the context of overall environment. This area therefore should be addressed prominently.

In mission critical applications, issue of data availability is also extremely important because if due to inevitable disaster or enemy attack the only storing and controlling entity is destroyed, the entire network will become dysfunctional. Availability of updated disaster information is extremely important for effective response and efficient actions.

The sensor data behaves differently in many ways from the data in traditional databases. WSNs need a mechanism to register processed queries and data dissemination. The data must be stored in a decentralized manner also to lower energy dissipation as well as faster data retrieval.

The scalability of the deployed architecture is another important issue. The hierarchical-clustering [23] has proved very useful for applications that require scalability to hundreds or thousands of nodes. Since the nature of the environment changes constantly, the sensor data has a temporal time interval in which they are valid therefore supporting energy-efficient real-time routing in WSNs is very challenging.

Last but not the least; the system should be able to work continuously even in adverse conditions. In crisis situations destruction of some critical components of the system is inevitable.
1.9 THESIS CONTRIBUTION

Most of the existing protocols are only suitable for specific types of applications and do not work well in large-scale applications. Some of the approaches improve energy efficiency by using clustering but they do not work efficiently when applied to large-scale sensor networks.

Although real time guarantee is provided by some protocols but they consumes lot of energy thus shortening the lifetime of the network. The data retrieval is also inferior and sluggish in existing protocols which can affect the mission critical applications. In existing protocols, the network becomes dysfunctional if the single controlling entity which stores long term data is destroyed due to any reason. The existing approaches are ineligible for handling situations pertaining to real world mission critical applications where data availability is crucial and rapid as well as automatic action is required.

A new energy efficient architectural framework for data placement and management is therefore proposed through this dissertation to effectively handle crisis situation. The proposed architecture handles the issues of data availability, data reliability, data integrity, fault tolerance, scalability besides providing real time guarantee as well as rapid, effective and automatic action in case of crisis.

In the proposed architecture, clustering has been used for enhancing network lifetime, reducing network contention and distribution of control over the network. The centralized clustering mechanism is used in the proposed architecture in case of small coverage area of the cell instead of relatively low energy sensor nodes deciding among themselves the new leader in many rounds of communication and spending precious energy. It will result in improved energy efficiency, well dispersion of CH nodes throughout the network and speedier clustering process since a single energy and resource rich entity takes a decision regarding the new leader.
However, in a large scale WSN if all the nodes submit their candidature to the single entity, it results in a huge expenditure of energy. In this dissertation to deal with the peculiarities of a large scale sensor network usage of multiple level of clustering within a cell has been proposed for minimizing total energy requirement and efficient communication of the sensed information over longer distances. The number of clustering levels and number of lower level clusters reporting to their parent CH will depend upon size of the cell according to the requirements of specific applications.

The cluster based hierarchical model used in the proposed architecture is also better than traditional multi-hop models. In cluster based hierarchical model, data travels from a lower clustered layer to a higher one. Although, it hops from one node to another but as it hops from one layer to another it covers large distances. The proposed architecture therefore achieves better deadline miss ratio, besides improving the energy efficiency.

Further, in the proposed architecture, for achieving energy efficient real time communication three different approaches have been used:

1. CHs of various clusters are changed after every round and remaining battery level is used as a major factor for electing CH.
2. CHs of various clusters are changed after every round but distance from controlling authority is used as a major eligibility factor for electing CH.
3. The node once chosen as CH continues in that role and it is changed only when its battery level falls below a threshold value.

The fault-tolerant operation is very critical for the success of WSN. The proposed architecture is based on the proven cellular technology. In the proposed architecture, an Action and Relay Station (ARS) lying on the shared edge between two adjacent cells replicates the concurrent information in the BSs of both the cells. A cell can have maximum of six neighbors thus we have used six ARSs per cell. The proposed architecture therefore will not become dysfunctional even if one or more ARSs or BSs within a cell stops working due to hostile conditions.
In the proposed architecture, it is assumed that all the nodes maintain information about the known ARSs and their status. The information is obtained when sensor node receives invitation messages for joining cluster broadcasted by various CHs from time to time that contains the location of the ultimate destination (i.e. ARS). Since each node knows its position in case of failure of the current destination due to any eventuality it sends the join request to the CH that is connected to the second nearest ARS.

The performance of proposed model in case of small coverage area has been compared with popular Centralized-Clustering algorithm LEACH-C [24] taking exactly same environment and assumptions. The number of rounds that first node and last node dies is used as a key indicator to evaluate the proposed system.

In case of large coverage area performance of the proposed model has been compared for network lifetime estimation with popular Hierarchical-Clustering algorithm HEED [25] taking exactly the same environment and assumptions, on various parameters like averages energy consumption per node, total messages received by BS and average energy consumption of nodes having BS within their sensing radius.

For checking effectiveness of proposed architecture in achieving better deadline miss ratio, it was compared with one of the most famous real time architecture - the RAP [26] taking same environment and assumptions in terms of messages received by BS, percentage of messages which have achieved their deadline and longevity of the network over various rounds of communication. The performance was judged by varying number of cluster formation characteristics like changing CH after every round, changing CH only after its battery level falls below a threshold value or changing CH election weight-age.

It is demonstrated that the proposed architecture is suitable for monitoring smaller as well larger coverage area. The proposed architecture is fault tolerant and achieves better deadline miss ratio besides improving network lifetime. It is suitable therefore for handling crisis situation in case of disaster or military scenario.
1.10 THESIS OUTLINE

Chapter I has given an overview of the WSNs as well as overview and scope of this research, the remainder of this thesis is organized into six additional chapters. This thesis begins by providing review of the existing work. In subsequent chapters the various aspects pertaining to the requirement, design, development and evaluation of the proposed distributed architecture for real time data placement using cellular framework are discussed. The Chapter II presents detailed review of some existing real time as well as clustering protocols and their characteristic comparison. In this chapter the WSAN are also explained. The Chapter III describes in detail the architectural framework and structures of the various components of the proposed model. The Chapter IV explains the model of the designed architecture. The Chapter V presents statistical estimates of the various parameters effecting working of various components of the model. The Chapter VI shows the performance of the proposed model. The Chapter VII concludes the research work and presents directions for future research in this area.
CHAPTER 2
REVIEW OF LITERATURE AND GENERAL CONSIDERATIONS

2.1 INTRODUCTION

The recent technological advances have lead to the emergence of wireless sensor and actor networks. The sensors gather information regarding an event and actors perform the appropriate actions. In case of emergency, the system should take automatic action on the basis of gathered information rather than waiting for manual intervention; therefore the role of actors is becoming very important. The requirement of real-time, efficient and fault tolerant communication is extremely important in emerging applications. Supporting real-time communication in sensor networks faces severe challenges due to their wireless nature, limited resource, low node reliability, distributed architecture and dynamic network topology. In emerging applications based on WSN therefore there is a tradeoff between energy efficiency and delay performance depending upon applications requirement.

2.2 WIRELESS SENSOR AND ACTOR NETWORKS (WSAN)

A WSAN [22] is a kind of heterogeneous WSN composed of a large number of sensors and a small number of resource rich actors. It is a network of nodes that cooperatively senses the environment and may control it, enabling interaction between people or computers and the surrounding environment.

Sensors are economical and low power devices with limited sensing, computation and wireless communication capabilities. Actors are equipped with better processing capabilities, longer transmission radius and larger battery energy. To perform distributed sensing and acting tasks, sensors gather information about the physical phenomenon, while actors perform appropriate actions on the basis of gathered information. The integrated architecture of WSANs is shown in Figure 2.1
In WSANs, collaborative operation of the sensors enables the *distributed sensing* of a physical phenomenon. After sensors detect an event that is occurring in the environment, the event data is processed and transmitted to the actors, which receives, processes and eventually reconstruct the event data. The process of establishing data paths between sensors and actors is referred to as *sensor-actor communication* [22]. Once the event has been detected, actors coordinate to re-construct it, to estimate its characteristics and make a collaborative decision on how to perform the action. This process is referred to as *actor-actor communication* [22]. The operation of WSANs in short can be considered as timely event detection, decision making and actuation.

![Fig. 2.1 An illustration of an integrated architecture of WSANs [22]](image)

The existing and potential applications of WSANs span a wide range including real-time target tracking, homeland security, battlefield surveillance and biological or chemical attack detection. For example, in fire detection applications, sensors can relay the exact origin and intensity of fire to water sprinkler actors so that the fire can be extinguished before it spreads. Similarly, motion and light sensors in a building can detect the presence of intruders and command cameras or other instrumentations to track them. Furthermore, sensors for structural health monitoring in airplanes or spaceships can drive instruments to timely take counter-measures against critical mechanical stress or structural faults.
2.3 SURVEY OF EXISTING WORK ON REAL TIME PROTOCOL IN WSN

Recent advances in wireless sensor networks have led to rapid development of real-time (RT) applications. The researchers have proposed some real time delivery schemes for WSNs. A comprehensive review of the challenges in providing real-time communication in sensor networks can be found in [27][28]. To the best of our knowledge, RAP [26] is the first RT routing approaches for WSNs. Admittedly, the SPEED protocol [29] is one of the most important RT routing protocols and has been an inspiration for many other RT routing protocols [30][31]. The salient features of SPEED and RAP protocols are as follows:

- They take into account remaining lifetime before a data report expires and remaining distance the report has to travel.
- The desired velocity is calculated depending upon distance from the destination and the lifetime of a packet.
- Both protocols assume that the packet will not miss its deadline if it travels along a straight line towards the destination with the specified speed.
- Both use greedy geographical forwarding, but differ in the exact mechanism used.

The existing real-time routing protocols that have been explored are as under:

2.3.1 RAP

Chenyang Lu et al. develop real-time architecture and protocols (RAP) based on velocity [26]. RAP provides service differentiation in the timeliness domain by velocity-monotonic classification of packets. In order to facilitate, delivery of a high velocity packet before a low velocity one, velocity of the packet is calculated and its priority is set on the basis of packet deadline and destination, in the velocity-monotonic order. The architecture of RAP is shown in Figure 2.2 [26]. Sensing and control applications interact with RAP through a set of Query/Event Service APIs. A Query/Event Service layer submits the query or event information of an area. The sensor-base communication is supported by a network stack including a transport-layer Location Addressed Protocol (LAP), a Geographic Forwarding (GF) routing protocol, a Velocity Monotonic (packet) Scheduling (VMS) layer and a prioritized MAC. RAP assumes the routing layer is aware of physical geography.
A router can determine the physical location of the destination relative to itself and forward the packet in general direction of the destination. The GF is highly scalable with regard to number of nodes, network diameter and rate of change in topology [32]. The proposed protocol also supports a multiple priority scheduling of packets using a VMS, which prioritizes the packets and schedules them on the basis of their required speed of transmission.

Fig. 2.2 RAP communication architecture [26].

RAP differentiates its service types by the “deadline” field of packets and their corresponding destinations. It computes the required transmission velocity of data packets in advance and assigns priority to different packets and then data packets queue up at nodes to wait for the service. Packets with higher priority will get the services earlier. In RAP, the required velocity is updated at each hop to reflect the urgency of the report. A node uses multiple FIFO queues with different priorities to schedule reports of varying degrees of urgency. Each queue accepts reports of velocities within a certain range.
RAP further adjusts the waiting and back-off times at MAC layer based on the priority of the report being transmitted, so that one with higher priority has greater probability of accessing the channel. Although RAP distinguishes among different services and data packets with higher priority get services first, it cannot guarantee end-to-end real time transmission. Its real-time transmission mechanism also cannot be adjusted dynamically to satisfy different real-time requirements.

2.3.2 SPEED

SPEED is a stateless, adaptive, location-based, real-time routing protocol, which can be effectively used if the location information is available in all sensor nodes and location updates can be delivered to the source sensors regularly. It bounds the end-to-end communication delay by enforcing a uniform communication speed at every hop in the network through a novel combination of feedback control and non-deterministic QoS aware geographic-forwarding [29].

SPEED first exchanges one-hop transmission delay with neighbours to find out the neighbour’s information. It then chooses a route according to local geographical data and transmission velocity information. It also evades big delay links and routing traps through reverse pressure routing changing mechanism. In SPEED, a node actively controls the data rate to avoid congestion by maintaining a relatively stable relay-speed to each neighbour. The node measures the delay of sending a report to each neighbour using exponential weighted moving average.

SPEED aims to reduce the end-to-end, deadline-miss ratio in a sensor network. It is a real-time communication protocol for sensor networks. It supports soft communication based on feedback control and stateless algorithms. SPEED also provides three types of real time communication services: unicast, multicast and any-cast. It utilizes geographic locations to make localized routing decisions. The routing module in SPEED is called Stateless Non-Deterministic Geographic forwarding (SNGF) and works with other four modules at the network layer, as shown in Figure 2.3 [29].
Fig. 2.3 Routing components of SPEED [29].

The beacon exchange mechanism is used to collect information about nodes and their location. Delay estimation at each node is made by calculating the elapsed time when an ACK is received from a neighbour as the response to a transmitted data packet. SNGF scheme selects nodes that will meet the speed requirement by estimating delay values. In case no such node can be found, the relay ratio of the nodes is calculated.

Neighbourhood Feedback Loop (NFL) module is responsible for providing relay ratio of a node, which is fed to the SNGF module. The relay ratio of a node is calculated by looking at the miss ratios of its neighbours that could not provide the desired speed. The packet is dropped if the relay ratio is less than a randomly generated number between 0 and 1. When a node fails to find a next hop node, the backpressure-rerouting module is finally used to prevent voids and to eliminate congestion by sending messages back to the source nodes so that they will pursue new routes.

In comparison to Dynamic Source Routing (DSR) [33] and Ad-hoc on-demand vector routing (AODV) [34], SPEED performs better in terms of end-to-end delay and miss ratio. SPEED reduces transmission energy consumption, control packet overhead and traffic distribution. It is also able to achieve load balancing in the network to a great extent. SPEED although is a successful real-time WSN routing protocol based on simple routing algorithm but it is not energy efficient. SPEED uses only one delay threshold overall to manage transmission of data packets at the highest transmission velocity. As a result, it cannot satisfy different requirements for transmission delay and cause huge energy consumption.

The SPEED protocol results in energy exhaustion of nodes quickly because it selects nodes having high transmission velocity without considering the remaining energy of nodes. For more realistic understanding of SPEED’s energy consumption, therefore there is a need
for comparing it to a routing protocol, which is energy-aware. The idea of per-flow reservation appears to be non-scalable in SPEED due to the highly dynamic links and route characteristics; hence SPEED might not be scalable well for large WSNs.

As an extension of SPEED, the FT-SPEED [35] is proposed to handle the void problem caused by high sensor failure probability in WSN. In FT-SPEED, avoid announce scheme is designed to prevent packets from reaching the void through other routing paths. It also introduces a void bypass scheme to route the packets around two sides of a void to guarantee that the packets are delivered rather than just being dropped.

2.3.3 MM-SPEED

Multi-path and Multi-SPEED Routing Protocol (MMSPEED) [30] an extension of SPEED is designed to support multiple communication speeds and provides differentiated reliability. A key feature of MMSPEED is that it addresses both real-time and reliability separately. Scheduling messages with deadlines focuses on the problem of providing timeliness guarantees for multi-hop transmissions in a real-time robotic sensor application. In such application, each message is associated with a deadline and may need to traverse multiple hops from the source to the destination.

Message deadlines are derived from validity of the accompanying sensor data and start time of the consuming task at the destination. The protocol reduces deadline misses by scheduling message based on their per-hop timeliness constraints, careful exploitation of spatial reuse of the wireless channel and explicitly avoiding collisions.

It supports a probabilistic QoS guarantee by provisioning QoS in two domains - timeliness and reliability. QoS differentiation in timeliness is provided through multiple network-wide packet delivery speed guarantees.

The scheme employs localized geographic packet forwarding augmented with dynamic compensation, which compensates for local decision inaccuracies as a packet travels towards
its destination. The intermediate nodes can lift speed level if they find that the packet may miss the delay deadline on current speed but may meet it at a higher level.

The MMSPEED protocol rests on a few important assumptions:

1. All nodes know their geographical location.
2. Location of the packet destination is known.
3. The underlying MAC protocol allows prioritizing between different classes at least stochastically.
4. Each speed level is mapped onto a MAC layer priority class.

In supporting service reliability, probabilistic multi-path forwarding is used to control number of delivery paths based on the required end-to-end reaching probability. In this scheme, each node in the network calculates the possible reliable forwarding probability value of each of its neighbours to a destination by using the packet loss rate at the MAC layer.

According to the required reliable probability of a packet, each node can forward multiple copies of it to a group of selected neighbours from the forwarding neighbour set to achieve the desired level of reliability. These mechanisms for QoS provisioning are realized in a localized way without global network information, which is desirable for scalability and adaptability to large scale dynamic sensor networks.

Although, MM-SPEED [30] does some improvements over SPEED and differentiates among different real-time levels, it also does not dynamically adjust routing paths according to node’s energy state. The both SPEED and MMSPEED have a common deficiency that they do not take into account the energy consumption metric.

2.3.4 RPAR

Real-time Power-Aware Routing protocol (RPAR) [36] varies from the previously mentioned protocols in many ways:
1. It is the only protocol that combines power control and real-time routing to support energy-efficient, real-time communication.
2. It allows the application to control the trade-off between energy utilization and communication delay by specifying packet deadlines.
3. It is designed to handle faulty links.
4. It utilizes a novel neighbourhood management mechanism that is more efficient than the periodic beacons scheme adopted by SPEED and MMSPEED.
5. It uses dynamic transmission power adjustment and routing decision in order to minimize miss ratios. The transmission power has a large impact on the delivery ratio as it improves wireless link quality and decreases the required number of transmissions to deliver a packet.

The authors also perform a set of experiments using XSM2 [37] motes to demonstrate that transmission power control may be an effective mechanism for controlling communication delays under the light workloads by improving link quality and reducing the number of transmissions needed to deliver a packet. A trade-off can be made between energy consumption and communication delay by specifying packet deadlines.

Moreover, a novel on-demand neighbourhood management mechanism is proposed to reduce energy consumption in contrast to periodic beacon exchanging scheme adopted by SPEED and MMSPEED. The neighbourhood manager is invoked only when there are no eligible forwarding choices in the neighbour table for forwarding a packet.

The simulations show that the forwarding policy and neighbourhood management of RPAR together can introduce significantly reduction in energy consumption with desired real-time guarantee. However, the reaction time of the neighbour discovery is a potential problem to the real-time performance. Moreover, transmitting a packet at a high power level has a side effect of decreasing throughput due to increased channel contention and interference.
2.3.5 DGR

Directional Geographical Routing protocol [38] has been proposed to address the problem of real-time video streaming over a WSN with constrained bandwidth and energy. DGR constructs an application-specific number of multiple disjointed paths for a Video-sensor Node (VN) to transmit parallel FEC-protected H.26L real-time video streams, over a bandwidth-limited and unreliable networking environment. To cater for the characteristics of video transmission, multipath routing is used in DGR to support the delivery of multiple flows instead of single-path routing scheme typically based on shortest path, which will drain the energy of the nodes along some path thus shortening the network life.

The DGR spreads the paths in all directions in the proximity of the source and sink nodes, which imply that packets along some paths are likely to be forwarded to a neighbour farther to the sink than the node itself. This is just for picking paths that do not interfere with each other. However, DGR relies on such assumption that the VNs take turns to send video streams to the sink, therefore at any instance only one of the VNs is actively sending video data to the sink. This assumption is somewhat unreasonable. The DGR therefore cannot be deployed in large scaled sensor networks.

2.3.6 ACM

Kawai et al. [39] proposed Assured Corridor Mechanism (ACM) based on the synchronization-based data gathering scheme for urgent information transmission in sensor networks. In it, nodes in a corridor are kept awake for fast transmission, while adjoining nodes are kept silent for fewer collisions. The proposed approach implies that reliability and latency of transmission of emergency packets can be improved at the cost of larger transmission delay of non-urgent information and depletion of battery of awake nodes. They developed a mechanism for handling different emergency packets through the use of multiple corridors and priorities.
2.3.7 Other Protocols


Akkaya and Younis [40] presented an energy-aware QoS routing protocol that could find energy-efficient path along which the end-to-end delay requirement can be met. The proposed protocol extends the routing approach in [41] and finds a least cost as well as delay-constrained path for real-time data considering nodes’ energy reserve, transmission energy and other communication parameters. Moreover, it maximizes the throughput for non-real-time data by adjusting the service rate for both real-time and non-real-time data at sensor nodes.

![Queueing model on a particular sensor node](image)

**Fig. 2.4 Queuing model on a particular sensor node [42].**

In order to provide both real-time and best possible traffic at the same time, a class-based queuing model is employed. The queuing model is depicted in Figure 2.4 [42]. There is a classifier at each node to divert real-time and non-real-time traffic to different priority queues according to the type of incoming packets. The bandwidth ratio ‘r’ (actually an initial value set by the gateway) specifies the amount of bandwidth to be dedicated to both the real-time and non-real-time traffic on a particular outgoing link in case of congestion.
The classes can borrow bandwidth from each other when one type of the traffic is non-existent or under the limits.

The protocol is based on a two-step strategy incorporating both link-based costs and end-to-end constraints. First of all the $k$-least cost paths are calculated by using an extended version of Dijkstra’s algorithm without considering the end-to-end delay. Secondly, one of the path from the candidate paths is determined that meets the end-to-end QoS requirements and also maximizes the throughput for non-real-time traffic.

The simulation results show that their proposed protocol consistently performs well with respect to real-time and energy metrics but it is not scalable well in large WSNs because the routing protocol is an extended version of Dijkstra’s algorithm. To support end-to-end guarantee, their approach however does not consider the delay that occurs due to channel access at the MAC. In addition, the $r$-value is initially set same for all the nodes, which does not provide adaptive bandwidth sharing for different links. The main drawback of this approach is that it does not support multiple priorities for the real-time traffic. The protocol is extended in [43] by assigning a different $r$-value for each node in order to achieve better utilization of links.

**b. Yuan et al.**

Yuan et al. [44] proposed an energy-efficient real-time routing protocol for WSNs based on SPEED. They put forward a novel concept of *Effective Transmission (ET)* that ensures that the forwarding candidates are not only nearer to the sink but also farther from the source node with respect to its preceding node. It can therefore limit the area of the candidate nodes and efficiently improve the transmission.

Moreover, they separated the whole path’s end-to-end delay guarantee into the sum of point-to-point *Constrained Equivalent Delay (CED)*. Each intermediate node can independently decide its next forwarding node according to the value of this link’s CED. It can greatly reduce the overhead and simplify the route discovery process because there is no need to calculate the sum of each link’s delay on the whole path.
c. Z. Khalid et al.

Z. Khalid et al. [45] proposed a real-time energy-aware routing strategy for WSNs based on *Logical Network Abridgement (LNA)*. The LNA procedure is capable of describing the intrinsic state of health of the overall network. The protocol considers two cost functions: first one is for time awareness and the other is for energy awareness. There is still a lot of research needed on the selection of parameter values and to understand the relationship among different parameters used in the cost functions. Their future plans include extending the routing protocol to allow gateway mobility.

d. Pothuri et al.

Pothuri et al. [46] designed a novel heuristic solution to find energy-efficient path for delay-constrained data in WSNs. The proposed hierarchical network architecture models the access delays caused by the MAC layer in conjunction with the new routing framework. The set of paths between source and sink nodes are identified and indexed in the order of increasing energy consumption. End-to-end delay is estimated along each of the ordered paths and the one with the lowest index that satisfies the delay constraint is selected. However, their solution is based on the assumption that nodes are equipped with two radios: a low-power radio for short-range and a high-power radio for long-range communication such that each node can reach the sink directly using its long-range radio. This requirement is energy inefficient and may not be practical.

e. Ergen et al.

Ergen et al. [47] presented an energy efficient routing algorithm with hard delay guarantee for sensor networks. The algorithm is based on the model that all data packets are destined for a single sink. It aims at maximizing the lifetime of a WSN by adjusting the number of packets traversing each other.

To achieve this goal, the authors first exclude the delay constraint and formulate lifetime maximization as a linear programming (LP) problem to determine optimal routing paths and implement a distributed mannered solution which uses an iterative algorithm to approximate
the centralized optimal one. The delay guarantee is then incorporated into the energy efficient routing by limiting the length of routing paths from each node to the sink. Simulations reveal that the lifetime increases significantly and the delay guarantee can be satisfied. However, as also mentioned in [48], generally the result is not flexible enough to meet application specified delay bound.

2.3.12 Summary of real time routing protocols

In the Table 2.1 given below, a comparative summary of the RT routing protocols [49] has been described based on their RT Types, hierarchical architecture, location based, scalability, energy efficiency and link reliability.

Table 2.1 Classification of RT routing protocols in sensor networks

<table>
<thead>
<tr>
<th>Routing Protocol</th>
<th>RT type</th>
<th>Hierarchical</th>
<th>Location based</th>
<th>Scalability</th>
<th>Energy efficiency</th>
<th>Link reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAP</td>
<td>SRT</td>
<td></td>
<td>✓</td>
<td>Good</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Akkaya et al.</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Low</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>SPEED</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Good</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MMSPEED</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Good</td>
<td>N/A</td>
<td>High</td>
</tr>
<tr>
<td>RPAR</td>
<td>SRT</td>
<td></td>
<td></td>
<td>Good</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Z. Khalid et al.</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Moderate</td>
<td>High</td>
<td>N/A</td>
</tr>
<tr>
<td>Yuan et al.</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Good</td>
<td>High</td>
<td>N/A</td>
</tr>
<tr>
<td>DGR</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Moderate</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
<tr>
<td>Pothuri et al.</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Moderate</td>
<td>Moderate</td>
<td>N/A</td>
</tr>
<tr>
<td>Ergen et al.</td>
<td>HRT</td>
<td></td>
<td></td>
<td>Moderate</td>
<td>High</td>
<td>N/A</td>
</tr>
<tr>
<td>ACM</td>
<td>SRT</td>
<td>✓</td>
<td></td>
<td>Moderate</td>
<td>N/A</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

SRT – Soft Real Time, HRT – Hard Real Time
2.4 SURVEY OF EXISTING WORK ON CLUSTERING

The nodes are often grouped together into disjoint and mostly non-overlapping clusters to minimize communication latency and improve energy efficiency. Leader of every cluster is often referred to as the cluster-head (CH).

A CH may be elected by the sensors in a cluster or pre-assigned by the network designer. A CH may also be just one of the sensors or a node that is richer in resources. The cluster membership may be fixed or variable. CHs may form a second tier network or may just ship the data to interested parties, e.g. a base-station or a command center. The clustering has numerous advantages as mentioned below:

1. It localizes the route set up within the cluster resulting in reduction of the size of the routing table stored at the individual nodes [42].
2. It conserves communication bandwidth since it avoids redundant exchange of messages among nodes [50].
3. It stabilizes network topology at the level of sensors and thus cuts on topology maintenance overhead. Sensors only communicate with their CHs [51].
4. It facilitates implementation of optimized management strategies to enhance the network operation and also to prolong battery life of the individual sensors.
5. It schedules activities in the cluster so that nodes can switch to the low-power sleep mode most of the time and reduces their rate of energy consumption.
6. It engages sensors in a round-robin order so that redundancy in coverage can be limited and medium access collision is prevented [52]-[53].
7. It decreases the number of relayed packets by aggregating the data collected by the individual sensors by using functions such as suppression (eliminating duplicates), min, max and average [54].

Following is the detail of various clustering algorithms for WSNs in terms of their objectives, features, complexity, location awareness and shortcomings etc.
2.4.1 Random Competition Based Clustering (RCC)

Although, RCC [55] is designed for mobile ad-hoc networks, it is also applicable to WSNs. RCC mainly focuses at cluster stability in order to support mobile nodes. The RCC algorithm applies the First Declaration Wins rule, in which any node can ‘‘govern’’ the rest of the nodes in its radio coverage if it is the first to claim being a CH.

After hearing the claim which is broadcasted by the first node, neighbouring nodes join its cluster as member and give up their right to be a CH. Periodically every CH in the network broadcast a CH claim packet to maintain clusters. Since there is a time delay between broadcasting a claim packet and receiving it, concurrent broadcast can possibly create a conflict. Being unaware of on-going claims, many neighbouring nodes may broadcast CH claim packets concurrently.

To avoid such a problem RCC explicitly employs a random timer and uses the node ID for arbitration. Each node in the network reset its random time value, every time before broadcasting its CH claim packet. During this random time if it receives a broadcast message carrying CH claim packet from another node, it simply ceases the transmission of its CH claim. Since random timer is not a complete solution, RCC resolve further the concurrent broadcast problems by using the node ID. If the conflict persists, node having lower ID will become the CH. A CH in adaptive clustering abandons its role when it hears a node with a lower ID, while, a CH in RCC only gives up its position when another CH moves near to it.

2.4.2 Banerjee and Khuller

The goal of Banerjee and Khuller is to form a multi-tier hierarchical clustering [56]. Figure 2.5 illustrate the concept of hierarchy of clusters. A number of cluster’s properties such as cluster size and the degree of overlap, which are useful for the management and scalability of the hierarchy, are also considered while grouping the nodes. In the proposed scheme, any node in the WSN can initiate the cluster formation process. Initiator with least node ID will take precedence, if multiple nodes started cluster formation process at the same time.
The algorithm proceeds in two phases: Tree discovery and Cluster formation. The tree discovery phase is basically a distributed formation of a Breadth-First-Search (BFS) tree rooted at the initiator node. Each node ‘u’ broadcast a signal once every ‘p’ units of time, carrying the information about its shortest hop-distance to the root ‘r’. A neighbouring node ‘v’ of ‘u’ will choose ‘u’ to be its parent and will update its hop-distance to the root if the route through ‘u’ is shorter. Broadcast signal carry the information such as source ID, parent ID, root ID and sub-tree size. Every node updates its sub-tree size when its children sub-tree size change.

The cluster formation phase starts when a sub-tree on a node crosses the size parameter, ‘k’. The node initiates cluster formation on its sub-tree. It will form a single cluster for the entire sub-tree if sub-tree size is $< 2k$, or else, it will form multiple clusters. It is crucial for clusters to keep cluster information after the cluster creation phase whereas maintenance of BFS tree is not so important.

2.4.3 GS$^3$

Zhang and Arora [57] present an algorithm, called GS$^3$ for self-configuring a wireless network into a cellular hexagon structure. They define the radius of the circle that contains all
nodes in the cluster as a measure for the geometric size. A large cluster radius is said to increase energy consumption, reliability for intra-cell communication and limit the spatial reuse of radio frequencies in the network. Two kinds of nodes are assumed in the system: big and small. The big nodes are responsible for initiating the cluster formation process. In addition they also act as mediator by interfacing the small nodes to other cells and network, e.g. the Internet.

It should be noted that the cellular hexagon structure is virtual and is just used to guide the grouping and redistribution of nodes. To form the cellular hexagon structure, the area is divided into cells of equal radius ‘R’. One of the big nodes starts the clustering process by selecting the heads of neighbouring cells which select their neighbours and so on. Unselected members become cell members. Upon their selection cell heads relocate to the centers of their cells and start establishing their neighbouring cells by selecting their heads. The process is repeated until no more cells could be added.

GS$^3$ differs from other distributed clustering mechanisms by guaranteeing a predictable placement and number of CHs in a system. Unlike the hierarchical clustering algorithm of [56] in which convergence under perturbations requires multiple rounds of messages, GS$^3$ offers one-way diffusion within perturbed areas. Since, GS$^3$ uses geographic radius of cluster instead of logical radius, long intra-cluster links are possible and it guarantees the logical radius of clusters implicitly.

2.4.4 Energy Efficient Hierarchical Clustering (EEHC)

Bandyopadhyay and Coyle [58] proposed EEHC which is a distributed, randomized clustering algorithm for WSNs with the objective of maximizing the network lifetime. CHs collected the sensors’ readings in their individual clusters and send an aggregated report to the base-station. Their technique is based on two stages - initial and extended.

In the initial stage, also called single-level clustering, each sensor node announces itself as a CH with probability ‘$p$’ to the neighbouring nodes within its communication range. These CHs are named as the volunteer CHs. All nodes that are within ‘$k$’ hops range of a CH
receive this announcement either by direct communication or by forwarding. Any node that receives such announcements and is not itself a CH becomes the member of the closest cluster. Forced CHs are nodes that are neither CH nor belong to a cluster. If the announcement does not reach to a node within a preset time interval ‘t’ that is calculated based on the duration for a packet to reach a node that is ‘k’ hops away, the node will become a forced CH assuming that it is not within ‘k’ hops of all volunteer CHs.

In the second stage, the process is extended to allow multi-level clustering, i.e. building ‘h’ levels of cluster hierarchy. Like [56], the clustering process is recursively repeated at the level of CHs to form an additional tier. The algorithm opts to ensure h-hop connectivity between CHs and the base-station. Assumed that level ‘h’ is highest, sensor nodes transmit the collected data to level-1 (lowest level) CHs. The CHs at the level-1 transmit the aggregated data to the level-2 CHs and so on. At the top level of the clustering hierarchy, CHs transmit the aggregated data report to the base station.

Energy consumption in various network operations like sensor data collection, transmission of aggregated information to base station etc. will depend on the parameters ‘p’ and ‘k’ of the algorithm. The authors have specified the mathematical expression for the values of ‘p’ and ‘k’ to achieve minimal energy consumption. The derivation is based on periodic generation and transmission of sensors data and employs stochastic geometry to estimate communication energy. Simulation results confirmed that by using optimal parameter values energy consumption in the network can be reduce significantly.

2.4.5 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [59] is one of the most popular algorithms for WSNs. It forms clusters based on the received signal strength and uses the CH nodes as routers to the base-station. All the data processing such as data fusion and aggregation are local to the cluster. It guarantees that every node evenly become CH, but does not take into account battery level and the interrelationship among nodes. In LEACH, CHs are determined in a distributed autonomous fashion. At each round $l$, each node $v$ independently decides to be a CH with probability $P_v(l)$ if the node $v$ has not been a CH in the most recent $\left( l \mod \left( \frac{N}{k}\right) \right)$ rounds [59].
\[ P_v(l) = \frac{k}{\frac{N}{k}(mod^{N})} \]  \hspace{1cm} (2.1)

where \( k \) is the average number of CHs for each round.

This means that each node becomes CH at least once every \( \frac{N}{k} \) rounds. In this algorithm a node with very low energy may be selected as a CH since the decision to elect the CH is probabilistic. The whole cluster becomes dysfunctional when this node dies. The CH is assumed to have a long communication range so that the data can reach the base-station from the CH directly. This is not always a realistic assumption since the CHs are regular sensors and the base-station is often not directly reachable to all nodes. LEACH also forms one-hop intra- and inter cluster topology where each node can transmit directly to the CH and thereafter to the base-station. Consequently, it is not applicable to networks deployed in large regions.

### 2.4.6 LEACH-C

LEACH-C [24] is a centralized cluster formation version of LEACH, where the BS organizes and controls the network. More precisely, LEACH-C protocol provides a centralized cluster formation, local processing for aggregation of sensed data and the rotation of CHs for every round. These activities are aimed at achieving uniform energy consumption among sensor nodes and maximizing network lifetime. Since, the BS does not have energy constraint, centralized cluster formation methods can be attractive alternatives. In LEACH-C, the cluster formation is formulated as a p-median problem [60], which is one of the well-known facility location problems. This algorithm produces better clusters by dispersing the cluster head nodes throughout the network.

### 2.4.7 Hybrid Energy-Efficient Distributed Clustering (HEED)

HEED [25] is a distributed clustering scheme in which CH nodes are picked from the deployed sensors. HEED considers a hybrid of energy and communication cost when selecting CHs. Unlike LEACH, it does not select cell-head nodes randomly. Only sensors that have a high residual energy can become cell-head nodes. HEED has three main characteristics:
1. The probability that two nodes within each other’s transmission range will become CH is small. Unlike LEACH, this means that CHs are well distributed in the network.
2. Energy consumption is not assumed to be uniform for all the nodes.
3. The probability of CH selection can be adjusted to ensure inter-CH connectivity for a given sensor’s transmission range.

In HEED, each node is mapped to exactly one cluster and can directly communicate with its CH. The algorithm is divided into three phases:

a. **Initialization phase**

   The algorithm first sets an initial percentage of CHs among all sensors. This percentage value, $C_{prob}$, is used to limit the initial CH announcements to the other sensors. Each sensor sets its probability of becoming a cluster-head, $CH_{prob}$, as follows [25]:

   $$CH_{prob} = \frac{E_{residual}}{E_{max}} \times C_{prob}$$

   where $E_{residual}$ is the current energy in the sensor and $E_{max}$ is the maximum energy, which corresponds to a fully charged battery. $CH_{prob}$ is not allowed to fall below a certain threshold $P_{min}$, which is selected to be inversely proportional to $E_{max}$.

b. **Repetition phase**

   During this phase, every sensor goes through several iterations until it finds the CH that it can transmit to using the least transmission power. The sensor elects itself to be a CH, if it does not hear from any CH and sends an announcement message to its neighbours informing them about the change of status. Finally, each sensor doubles its $CH_{prob}$ value and goes to the next iteration of this phase. It stops executing this phase when its $CH_{prob}$ reaches 1. There are therefore two types of cell-head status that a sensor could announce to its neighbours:

   - Tentative status: The sensor becomes a tentative CH if its $CH_{prob}$ is less than 1. It can change its status to regular at a later iteration if it finds a lower cost CH.
   - Final status: The sensor permanently becomes a CH if its $CH_{prob}$ has reached 1.

   c. **Finalization phase**

   During this phase, each sensor makes a final decision on its status. It either picks the least cost CH or pronounces itself as CH.
In HEED lifetime is prolonged by reducing the number of nodes that compete for channel access and routing information through an overlay among Cluster Heads. HEED improves network lifetime over generalized LEACH because generalized LEACH randomly selects Cluster Heads resulting in faster death of some nodes. It uses less energy in clustering than generalized LEACH because it does not propagate residual energy information.

2.4.8 Summary of clustering protocols

A comparative summary of the clustering protocols [61] described above is given in Table 2.2 based on cluster-count, intra and inter cluster topology, node type, methodology, CH-selection and algorithm complexity.

### Table 2.2 Classification of clustering protocols in sensor networks

<table>
<thead>
<tr>
<th>Clustering Protocol</th>
<th>Cluster Count</th>
<th>Intra-Cluster Topology</th>
<th>Inter-Cluster Topology</th>
<th>Node Type</th>
<th>Methodology</th>
<th>CH Selection</th>
<th>Algorithm Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCC</td>
<td>Variable</td>
<td>Adaptive</td>
<td>Direct Link</td>
<td>Sensor</td>
<td>Hybrid</td>
<td>Random</td>
<td>Variable</td>
</tr>
<tr>
<td>Baneerjee &amp; Khuller</td>
<td>Variable</td>
<td>Adaptive</td>
<td>Multi-Hop Hierarchical</td>
<td>Sensor</td>
<td>Distributed</td>
<td>Random</td>
<td>Variable</td>
</tr>
<tr>
<td>GS³</td>
<td>Preset</td>
<td>Adaptive</td>
<td>Direct Link</td>
<td>Resource</td>
<td>Distributed</td>
<td>Pre-assigned</td>
<td>Variable</td>
</tr>
<tr>
<td>EEHC</td>
<td>Variable</td>
<td>Adaptive</td>
<td>Direct Link/Hierarchical</td>
<td>Sensor</td>
<td>Distributed</td>
<td>Random</td>
<td>Variable</td>
</tr>
<tr>
<td>LEACH</td>
<td>Variable</td>
<td>Fixed (1-Hop)</td>
<td>Direct Link</td>
<td>Sensor</td>
<td>Hybrid</td>
<td>Random</td>
<td>Constant</td>
</tr>
<tr>
<td>Leach-C</td>
<td>Variable</td>
<td>Fixed (1-Hop)</td>
<td>Direct Link</td>
<td>Sensor</td>
<td>Centralized</td>
<td>Random</td>
<td>Constant</td>
</tr>
<tr>
<td>HEED</td>
<td>Variable</td>
<td>Fixed (1-Hop)</td>
<td>Multi-Hop Hierarchical</td>
<td>Sensor</td>
<td>Distributed</td>
<td>Random</td>
<td>Constant</td>
</tr>
</tbody>
</table>
2.5 CONCLUSION

In this chapter, the WSANs were explained and detailed reviews of some existing real time as well as clustering protocols were given. The WSANs cooperatively sense the environment and may control it. The actors collect and process sensor data and perform appropriate actions on the environment. Observing this, the researchers have proposed some real time delivery schemes for WSNs. The various real time protocols have also been compared in this chapter on the basis of location, the hierarchical architecture, scalability, energy efficiency and link reliability. Keeping the goal of energy conservation, the study of existing clustering protocols and their comparison on the basis of cluster count, inter as well as intra cluster connectivity, CH selection and methodology adopted for choosing CH is peer reviewed and given in detail.
CHAPTER 3
DISTRIBUTED CELLULAR ARCHITECTURE FOR DATA PLACEMENT IN WIRELESS SENSOR NETWORKS

3.1 INTRODUCTION

The new WSN based architectural framework for data placement and management has been proposed for providing the capability to monitor the environment and to take automatic as well as effective action in crisis situations, on the basis of available information of the critical area. In the proposed architecture each sensor monitors, collects and generates data stream about its environment. It is assumed that the sensor nodes are aware of their locations and are time synchronized. The proposed architecture is based on cellular network framework. The proposed architecture has been designed to help in faster decision making and to take comprehensive as well as automatic action on the basis of it.

The proposed architecture is based on a cellular framework. In it for reducing energy consumption clustering has been used. The entire data of a cell is stored in a base station located at the centre of the cell and for data dissemination and actions, action and relay stations (ARSs) have been introduced. In order to reduce energy consumption instead of communicating with all sensor nodes, the ARS communicates only with CHs reporting to it. The data is distributed over the sensor network by routing from cluster heads to the nearest ARS. The architecture is expected to provide various features like timeliness guarantee, fault-tolerance, data integrity and data-centric as well as distributed storage besides enhanced energy efficiency.

3.2 ARCHITECTURAL FRAMEWORK

The proposed architecture based on cellular framework is fault tolerant, real-time and distributed. It supports basic kind of data integrity and disaster management approaches for WSNs [22, 62]. The main design paradigms in the architectural framework are explained below:
3.2.1 Cluster management

One of the crucial challenges in the organization of sensor networks is energy efficiency because battery capacities of sensor nodes are severely limited and replacing the batteries is not practical. The radio communication that includes both data transmission and reception consumes a significant portion of the energy of a sensor node. The energy efficient communication and data gathering mechanisms therefore are key issues that have to be considered to prolong the lifetime of a WSN.

The clustering has proven to be an effective approach for enhancing energy efficiency besides organizing the network into a connected hierarchy. It divides sensors into groups (clusters), so that sensors communicate information only to cluster-heads (CH) and then the cluster-heads sends the aggregated information towards the destination thereby saving precious energy. Cluster-heads are responsible for coordination among the nodes within their clusters. The operation of cluster based sensor network is divided into rounds. Each round has two phases - clustering phase and data transmission phase. Rounds are repeated to monitor events continuously.

Fig. 3.1 Clustering for Small Coverage Area of Cell
In the proposed architecture, depending upon the area of ambient environment to be monitored two types of clustering mechanisms has been used for increasing the lifetime of the network. The centralized clustering approach and one hop communication has been used whenever area to be monitored is small, as shown in Figure 3.1. The hierarchical clustering mechanism and multi hop communication has been used, for handling the peculiarities of large area sensor network, as depicted in Figure 3.2. In case of large coverage area of the cell, use of centralized clustering mechanism will result in a huge expenditure of energy.

![Diagram of Clustering for Large Coverage Area of Cell](image)

**Fig. 3.2 Clustering for Large Coverage Area of Cell**

The centralized clustering process is illustrated in Figure 3.3. In this mechanism, all alive nodes of a cluster that have battery level greater than the minimum battery level required to operate, submits their candidature for becoming new CH via current CH to their controlling authority, ARS. The ARS then chooses CHs for the next round from the submitted candidatures on the basis of remaining battery level. This process saves the precious energy of sensor nodes, which otherwise they have to spend in deciding among themselves new CHs in many rounds of communication.
In the proposed architecture, in order to further reduce energy consumption instead of communicating with all sensor nodes, the ARS communicates only with CHs reporting to it. The CHs submits candidature of all members of their own clusters for becoming CH in next round to their respective ARS’s. It results in faster completion of the clustering process and minimum use of invaluable energy of nodes. The aim of ARS is to determine the best possible candidates for becoming CHs in next round. Only the sensor nodes having minimum operating battery level are eligible for becoming CH. In order to further reduce energy consumption and to choose well distributed CHs from the eligible sensor nodes to minimize the average distance each sensor node has to traverse to send its data, the ARS may use the solution of p-median problem specified in [60]. The centralized clustering algorithm has been specified in Table 6.2.

![Fig. 3.3 Centralized Clustering Process](image)

As specified in Figure 3.2 in the proposed architecture, for large scale sensor networks [9][26] hierarchical clustering mechanism has been used to improve energy efficiency and longevity of WSN. In hierarchical clustering, level $k$ CH aggregates and sends the data received from its members to higher level $k - 1$ CH. The level $k - 1$ CH sends data received from lower level clusters as well as the aggregated data of sensors attached to it to its parent CH. This process continues until the sensed data reaches the level 0 CH which finally relays it to its respective ARS. The number of clustering levels and number of lower level clusters reporting to their parent CH depends upon the size of cell as well as requirements of the specific application. In the proposed architecture, various clusters are formed in a top-down
manner. The detailed hierarchical clustering algorithm for selection of sensor node as CH has been explained in Table 6.4.

At regular intervals each member node of the cluster sends monitored data to its CH. Since data monitored by sensor nodes within a cluster are often correlated or redundant, entire data is not required. The CH therefore periodically carries out data aggregation process on the data received from the sensor nodes within the cluster to remove redundant data. It saves precious energy and network bandwidth since amount of data to be sent is reduced. In a cluster, CH consumes more energy than normal sensor nodes due to its additional responsibilities like receiving data from its cluster members, aggregating received data and sending aggregated data towards the destination. As a result CHs tend to die faster than normal nodes.

The CH role is rotated among nodes in a WSN to evenly distribute the burden carried by a CH among all nodes, thus giving an opportunity for all nodes to have approximately the same lifetime. The network life time can be improved further by selecting proper points at which a CH role should be relinquished to higher energy nodes via a CH rotation phase. Determining an optimal point at which the CH rotation should be carried out is very important. If a CH selection phase is triggered with a smaller number of data transmission rounds, it may result in excessive overhead during the CH selection phase. On the other hand, if the number of data transmission rounds is large before a CH selection is triggered, the CH nodes would not have enough energy to act as ordinary sensor nodes after relinquishing the CH role. Approach specified in [63] can be used for determining optimal point at which the CH rotation should be carried out.

3.2.2 Sensor-ARS coordination

In order to provide effective sensing and acting, coordination mechanisms are required among sensors and ARS. The sensors detecting a phenomenon transmit their data via CHs to the ARS which processes all incoming data and transmit it to base station(s) where it is stored. Depending upon urgency, the ARS can also initiate automatic action after obtaining the integrated data from the base station. It can also collaborate with other ARSs to have a
comprehensive knowledge about the critical area. The ARSs are resource rich nodes with high transmission power; therefore ARS-ARS communication can be long range.

3.2.3 Node placement

The sensor data is highly spatial in its nature therefore in order to interpret data in a meaningful way it must be correlated with its location. In sensor networks therefore location is more important than a specific node’s ID. The node placement in harsh isolated environments like target tracking and surveillance must be flexible, low-cost, durable and self-sustaining over long periods of time. Moreover, multiple sensors of several types should be deployed in the monitored area for measuring different environmental conditions like temperature, pressure, humidity etc.

In the proposed architecture, due to various constraints pertaining to the cost and size of sensors, energy consumption, ambient environment and the deployment of sensors, some assumptions regarding location information of its components have been made. The BS and ARS are assumed to have known location information and their locations are obtained by using Global Positioning System (GPS). The sensor nodes obtain their location information by using polar coordinates \((r, \theta)\) with respect to their respective ARSs, as the origin of the coordinate system \((0, 0)\).

3.2.4 Localization methods

There are four main methods that can be used to determine sensor node location. These methods are described as under:

i. Zenon Chaczko, Ryszard Klempous et al. [64]

The sensor node position \(P(x, y)\) can be found on 2D area if its distances from three points \((x_1, y_1)\), \((x_2, y_2)\) and \((x_3, y_3)\) are known. As shown in Figure 3.4, if there are three known points \(A(x_1, y_1)\), \(B(x_2, y_2)\), \(C(x_3, y_3)\) and if point \(P(x, y)\) is at a distance \(d_1, d_2, d_3\) respectively from these points, \(x\) and \(y\) can be determined. Methods of estimating distances are generally related to circumstances of signal propagation, which depends upon two main parameters attenuation and delay.
Fig. 3.4 Finding point’s position knowing distance to three known points. [64]

In this method if signal with power $P_0$ is sent at $t_0$, then the received signal power $P_1$ is

$$P_1 = \frac{P_0}{d_1^m}$$  \hspace{1cm} (3.1)

and is delayed by $t_1$ where

$$t_1 = \frac{d_1}{v}$$  \hspace{1cm} (3.2)

In equations (3.1) and (3.2) $d_1$ is a distance between two points, $m$ is a constant that depends on environment and $v$ is a speed of signal propagation in the environment. Both the parameters therefore are used to estimate distance $d_1$.

ii. Time of Arrival (ToA) Algorithm [65]-[68]

This method is based on signal delay. If sensor $P$ with unknown position $(x, y)$ sends signal $s(t)$ the received signals can be given by equation (3.3)

$$y_j(t) = ks(t - t_j)$$  \hspace{1cm} (3.3)

where $j = 1,2,3$ refer to receivers located in known positions $(x_j, y_j)$. Assuming a perfect synchronization, distances between transmitter $P$ and receivers can be easily computed.

iii. Time Difference of Arrival (TDoA) Algorithm [69, 70]

In this method when transmitter $P$ sends a message and the received signal correlation (delay) in more than two receivers, the distances between point $P$ (whose position has to be determined) and each receiver (whose positions are known) can be computed.
iv. **Receive Signal Strength (RSS) Algorithm [67, 71]**

Unlike ToA and TDoA, RSS algorithm is not based on signal delay but on signal strength analysis. If the transmitted signal strength $P_0$ is known, distance between sensors can be estimated by measuring received signal strength $P_1$ as follows:

$$d_1 = m \sqrt{\frac{P_0}{P_1}} \quad (3.4)$$

It is assumed that signal strength decreases with the distance. Disadvantage of this method is necessity of estimating constant $m$ that depends on environmental conditions.

**3.2.5 Addressing scheme**

The addressing scheme in traditional networks is fixed x-y coordinate address. In the proposed architecture, addressing format is $<$Location ID, Node Type ID$. The Location ID identifies the location of a node that conducts sensing activities in a specified region of the network. The Node Type ID describes the functionality of the node. The Location ID can be used to make localized routing decisions. In a pure localized algorithm, action invoked by a node should not affect the system as a whole.

**3.2.6 Data routing**

In the proposed architecture, in order to reduce energy consumption, a sensor node sends the sensed data to its respective CH. The clusters head then routes the aggregated data towards the pertaining ARS using single hop or multi hop communication. The ARS ultimately sends the received data to the base station. The ARS checks the current status of the base station(s) before sending data to it. The data is not sent to a BS if its current status is collapsed; so that precious bandwidth is not wasted.

Gyan Ranjan, Amit Kumar et al. [72] proposed a method based on location awareness to optimize route discovery by minimizing energy consumption. If $Q$ is the minimum energy required to transmit a packet across a zone, a head node while initiating route discovery can broadcast a packet with an energy $E(Q \leq E < 2Q)$ to ensure that response is received from its neighbouring clusters. A receiving head in the neighbouring area responds by sending its polar coordinates $(r, \theta)$ to the initiating head.
3.2.7 Data storage

In WSN, the individual identity of nodes is always insignificant. In the proposed architecture for storing information, as suggested in [73], [74] and [75] the concept of data centric storage has been used. In data centric storage, data can be stored and retrieved by name therefore the communication primitives are organized around the sensed data instead of network nodes which is normally the case in wired or conventional wireless networks. It enables nodes within the network to store or cache the data transparently. In [76] there is a novel idea for storing data at a node, which has been generated at another node according to the name of data.

Further, the data centric storage also enables use of indices or keys for efficient access to data in large-scale sensor networks. Using data centric storage mechanism in the proposed architecture, cost of accessing events is almost zero since all the events are available at one node (base station). To further speed up the storage and retrieval of data, hashing mechanism has been used.

![Cellular Network Framework](image)

Fig. 3.5 Cellular Network Framework.
3.2.8 Distributed storage and replication

One of the aims of the proposed architecture is to prevent collapse of the communication system. Hence, the ARS lying on each pair of shared edges along the border between two cells passes the observations received from various CHs to the base stations of both cells. So, that in case of collapse of one base station the requisite information can be obtained from the adjacent cell’s base station. In case the base station receives redundant information it is ignored by it.

The Figure 3.5 shows the structure of a sensor network where base stations are in the centre and ARSs are at the boundary of the cell. In the proposed architecture relevant data is stored in various BSs in a distributed fashion. The distributed architecture results in speedier storage and query processing as well as enhanced availability. In the proposed architecture collapse of a BS may result in the loss of some of the overall system information only.

3.2.9 Retention period

The voluminous data can overload the limited storage and communication capacity of a sensor network. It is impractical to store a large volume of raw data locally at the data sources (BSs) or to transmit the data over the sensor network to a central depository (Sink) [75, 77]. The nature of the environment changes constantly therefore the sensor data has a temporal time interval in which they are valid. For example, temperature data is irrelevant after a certain time. The retention period for the data to be stored in the BS therefore is a very important parameter that controls the amount of data to be stored locally. The stored data is removed from the BS after expiry of its retention period to reduce the burden on BS’s memory. Some compressing techniques can also be used to further reduce the memory requirements to store the data.

3.2.10 Data aggregation

Sensor networks consist of high density of sensor nodes therefore multiple nodes may generate and transmit redundant data causing unnecessary energy consumption and hence lesser network lifetime. The energy consumption in a sensor node is mainly due to its three activities: data sensing, data processing and data transmission/reception.
Out of these three activities data transmission or reception consumes most energy. The amount of communication should be reduced by eliminating or aggregating the redundant sensed data to prolong the lifetime of the WSNs. Substantial work has been done in this area [78]–[81] to ensure efficient usage of communication resources that includes the low-bandwidth links.

3.2.11 Data integrity

The trustworthiness of information is extremely important to take an effective action. In the WSN only the summarized information is generally stored to increase the longevity of the WSN. Instead of storing information in individual CHs as suggested in some protocols, in the proposed architecture, storage of information of all clusters within a cell is suggested inside the corresponding base station. This will improve faithfulness of the information as the decision will be made taking into account broad number of facts. The information retrieval will also be faster because entire information will be available at a single place.

3.2.12 Defining a query

The approach suggested in [82] has been used for interaction with a sensor network as a relational database. Defining a query for the query processor means defining what activities must be carried out by each sensor in the network. The query in an SQL-like language MW-SQL [82] can be specified. A query is represented as a combination of operators of the query algebra connected by the data streams. The MW-SQL queries are expressed through query statements having the form:

```
SELECT select-list
FROM source
[ WHERE condition ]
[ EPOCH samples [ SAMPLES ] ]
[ EVERY rate ]
```

where keywords are represented in boldface.
3.2.13 Query processing

In WSN based applications, queries are processed based on sensed events therefore knowledge of the location of the event and its time of occurrence are important parameters. In addition to the sensed data attributes therefore a sensor tuple consists of the following attributes also to uniquely identify any sensed data tuple generated in the network:

- **NodeId**: NodeId represents geographical location of the source node that generates the data tuple.
- **TimeStamp**: TimeStamp represents the time at which the tuple was produced.

3.2.14 Query optimization

Query optimization is the process of selecting the most efficient query-evaluation plan from the many strategies usually possible for processing a given query. The difference in cost in terms of evaluation time between a good strategy and a bad strategy is often significant and may be of several orders of magnitude. Hence, it is worthwhile for the system to spend considerable time on the selection of a good strategy for processing a query, even if the query is executed only once. Various aspects for selecting an efficient strategy for processing the received query can be choosing the right algorithm to use for executing an operation, choosing the specific indices to use and so on.

In the proposed architecture, the BS processes the obtained data and sends the results to ARS using appropriate algorithm. It uses various optimization techniques for query processing and sending the results. The BS sends filtered or average of the sensed data obtained as a query result instead of sending large amount of raw records over the network to the ARS. The BS also ignores results that are outside some predefined threshold limits as specified by the sink. This could depend upon specific application requirements; for example, obtaining the temperature of a particular region.

3.2.15 Quality of service

Quality of Service (QoS) [83] provisioning is becoming a critical issue in designing WSNs. Providing QoS in WSNs is challenging, because it leads to a significant amount of computation and communication overhead, especially to the sensor nodes with limited
battery energy and poor computation ability. The transmission of sensed data should ensure efficient energy usage and meet applications requirements. The proposed architecture uses various existing QoS techniques [84]-[85] to optimize the usage of the network.

As an example, to handle periodic queries over dynamic data streams, in [84], a prediction based Quality-of-Service (QoS) management scheme has been explained, which predict the query workload using execution time profiling and input data sampling and adjusts the query QoS levels based on online query execution time prediction. The [85] discusses some key design considerations in providing QoS support.

![Fig. 3.6 Proposed System Architecture](image)

### 3.3 SYSTEM STRUCTURE

The proposed architecture is portioned into modules, each of which deals with various responsibilities of the overall system. Figure 3.6 shows these components and the connections among them. The basic components and their functionalities used in the design are as follows:
3.3.1 Sensor node

The sensor nodes are economical and tiny devices having several resource constraints in terms of energy, memory, communication and computation capabilities. Inspite of these resource constraints thousands of them may collectively monitor the physical world, disseminate information upon critical environmental events and process the received information. They are generally organized into clusters to reduce energy consumption and to increase lifetime of the network. A sensor node consists of the following four major components:

a. **Processing Unit**
   The processing unit is equipped with a processor and a small memory. Processor is responsible for control of the sensors and execution of communication protocols. Memory is present for storage of data for various purposes like data aggregation. The processing unit performs two main functions:
   - Supervising and controlling the working of other components
   - Collaboration with other nodes to accomplish assigned sensing task.

b. **Sensors and analog-to-digital converters**
   The sensors gather data from the environment and send it to the processor. They observe the phenomenon such as thermal, optics or acoustics event. In case of analog sensors, the signals are sent to the analog-to-digital converters that convert them to digital format.

c. **Transceiver unit.**
   The transceiver unit transmits and receives radio or optical signals. It also connects the node with the network.

d. **Power unit**
   It consists of batteries or solar cells.

Additional components such as power generator, global positioning system (GPS) and a location finding unit can also be attached with the sensor node depending upon the application specific requirements. Each sensor node can have Node type ID that describes the functionality of the node. In a two-dimensional plane, a node having sensing range $r_s$ can sense events in a circular coverage area. The behavior of a sensor node can be explained by considering the state machine as shown in Figure 3.7.
**SLEEP STATE:** The node turns off its radio and keeps Low Power Mode (LPM) and starts a timer whose duration is an exponentially distributed random variable. When the timer expires, the node goes to the WAKE UP STATE.

![State Transition Diagram]

**WAKE UP STATE:** The node turns its beacon channel on and wakes up from LPM and broadcasts a beacon message containing the channel condition. After the node sends a beacon message, its radio is converted to the data channel and it is ready to receive a data message. The node goes to the IDLE LISTEN STATE.

**IDLE LISTEN STATE:** The node starts a timer of a fixed active time that must be long enough to completely receive a packet. If a data message is received, the active timer is discarded and the node starts a wait timer to receive a beacon that is of a longer time than the active timer. The node renew the traffic rate along with intensity parameter from the received data message and goes to the ACTIVE state. Furthermore, its radio channel is switched from data channel to the beacon channel. Otherwise if the active timer expires before any packet is received, the node goes to the CALCULATE STATE.
**ACTIVE STATE:** After a node receives the first beacon coming from a node in the next cluster within wait timer, it calculates the cluster ID and the size of next cluster using received beacon message containing the channel condition and data message containing the traffic rate in network. Next cluster is the region containing node that has generated a beacon to receive data message. After the node computes the next cluster size, node goes to WAIT STATE instead of directly transmitting the data message. Otherwise, if the beacon waiting timer expires before receiving any beacon message, the node goes to the CALCULATE STATE.

**WAIT STATE:** Node starts a back-off time before transmitting a data message and its radio is switched to the data channel. The back-off time is a uniformly distributed random variable within 0 to a maximum value. If the node listens a data message whose sequence number is same with own data message within a back-off time, the node discards the own data message and goes to CALCULATE STATE to avoid duplicate packets. Otherwise, if the back-off timer expires, the node transmits the data message and goes to CALCULATE STATE.

**CALCULATE STATE:** The node calculates the next sleeping time from the intensity parameter $i$ using an exponentially distributed random function. After the node generates the random variable for sleeping time, the node goes back to the SLEEP STATE.

### 3.3.2 Base Station

The persistence of data is the primary concern, therefore the data being sensed by the nodes in the network is ultimately transmitted to a control centre or base station so that it can be stored and referred in the future. The base stations are responsible for data storage in a distributed cellular framework. Base stations have enhanced capabilities over simple sensor nodes since they have to do complex data processing. The storing of entire information of the cell in the base station results in better data integrity thereby ARSs need not spend time on processing data before taking an effective decision.

The functional components of the base station are broadly divided into storage manager component, query processor component, transceiver unit and power unit. The storage
manager component provides interface between the low level data stored in the database and queries submitted to the system. Its main goal is to simplify and facilitate access to data. The storage manager component may include authorization and integrity manager, file manager, buffer manager etc. The query manager component handles queries received from ARSs; it may further include DML compiler, metadata manager, query evaluation engine etc.

3.3.3 Action and Relay Station

The ARS nodes are placed on the bordering areas of cells and are responsible for data dissemination in a time efficient manner. They are resource rich nodes equipped with better processing capabilities, higher transmission powers and longer battery life. An ARS unit consists of six basic components – actuation unit, processing unit, controller (decision unit), data writer, power unit and transceiver unit. The decision unit functions as an entity that takes sensor readings as input and generates action commands as output. These action commands are then converted to analog signals and transformed into actions via the actuation unit. The ARS nodes are placed on each pair of shared edges along the border between two cells.

Every ARS supports two types of interfaces: ad-hoc relay interface and cellular interface. The ARS communicates with other ARSs and CHs by using ad-hoc interface. It uses cellular interface to communicate with base stations of cellular network. During disaster some ARSs may be collapsed but the network will not become dysfunctional since only one ARS is enough to convey data from sensor network of a cell to a base station. The “Data Writer” component of ARS can pass the information received from various CHs of WSN immediately to the corresponding base stations or it can store data received in its local buffer and after some time send the aggregated data to further conserve power as well as communication bandwidth.

3.3.4 Sink

The sink acts as a bridge between the WSN and the physical world. It supervises and synchronizes the working of various components of the proposed architecture. It reports the critical findings so that in case of some disaster or any other eventuality, an immediate help
from outside can reach the reported event area quickly. It monitors the entire system and depending upon the received feedback sets the value of different parameters of various components of the proposed architecture like retention period so that they can perform their work in an efficient manner. It also specifies the positions of non-functioning sensor nodes, ARSs or base station to the controlling authority so that they can be replaced.

3.4 CONCLUSION

The proposed architecture is based on WSAN; it not only provides real time information but also automatic action on the basis of it. The architecture is portioned into modules each of which deals with various responsibilities of the overall system. The basic components and their functionalities that are used in the design are:

a. Sensors nodes which are low-cost and low-power devices with limited sensing, computation and wireless communication capabilities.

b. The base stations responsible for data storage in a decentralized real time database framework.

c. The Action and Relay Stations (ARS) for initiating query processing and taking actions on the basis of query result.

d. The sink to supervise and synchronize the working of various components of WSN.

In the proposed framework, the base station is in the center and the ARS is at the boundary of the cell. In the proposed architecture storing of information of all clusters within a cell is suggested inside the BS in a distributed cellular framework. The usage of Action and Relay Stations (ARS) has been proposed, for data dissemination and action in the proposed architecture. To reduce average energy dissipation of sensor nodes, the data is sent to the nearest ARS rather than base station. The data is distributed over the sensor network by keeping routing from cluster heads to the nearest ARS.

Further, in the proposed architecture analysis of stored data can be done to prevent disasters or plan rescue operations in a better way. The damage of some components due to enemy attack or disaster may result in the loss of the fraction of overall system information but not complete collapse of the proposed system.
CHAPTER 4

MODEL OF THE DESIGNED ARCHITECTURE

4.1 INTRODUCTION

Model checking [86] is a fully automated technique for verification of the finite state systems. The user gives a description of the system and defines its requirements. Model checking is based on the idea of exhaustive exploration of reachable state space of a system model. A simplified model of the system can be built that preserves its underlying design characteristics and avoids known sources of complexity, when the system itself cannot be verified exhaustively. This technique is especially useful for reactive and distributed systems that are characterized by many interactions among processes. A model checker explores all states reachable from an initial state.

The basic unit of observation in a system is commonly known as a state. The reliable and correct behavior of sensor network based on infrastructure can be critical for human life, environment, decision-making, etc. and consequence of misbehavior can be catastrophic. A WSN can be treated as a massively distributed and deeply embedded system [87]. A WSN therefore can be described by a set of concurrent communicating processes in Process Meta Language (PROMELA) [88], so that it can be considered in terms of communicating finite state machines. The SPIN [89]-[91] is a model checking method and appropriate tool for verifying the sensor networks specifications.

PROMELA was developed by Gerard J. Holzmann in the AT&T Bell Laboratories. He also wrote the corresponding tool called “SPIN” (Simple PROMELA Interpreter), which he put in the public domain. SPIN is one of the most popular model checking tools. In April 2002 the tool was awarded the prestigious System Software Award for 2001 by the ACM [96]. In SPIN, the system models are described in a modeling language called PROMELA. The user needs to specify a high-level model of a concurrent system or distributed algorithm, in PROMELA.
Modeling and analysis of sensor networks require their formal specification [92]. Since, WSNs can be rather complex, their properties and environment to model must be chosen. In this chapter, models of different components of the architecture have been specified in PROMELA using a GUI based tool VIP.

Model checking is an effective tool for determining whether the requirements are consistent and correct with the intended behavior of the system, and also whether the system's design correctly implements the requirements. VIP tool is used in the present work to generate PROMELA code of the proposed architecture. The resulting PROMELA code can be model checked using SPIN to verify the system behaviour.

4.2 FORMAL DESCRIPTION TECHNIQUES

The formal description techniques (FDT’s) are techniques used by designers of software to ameliorate the quality of their products. Usually, without using formal methods, the software development process looks as shown in Figure 4.1. The problem is given to the developer as an informal specification. This specification very often is written as a technical document which may very well be interpreted and often does not cover important questions that are discovered during the development process. The task of the developer is to find a solution to this specification and to develop a structure for this solution.
The common way of application of formal techniques in order to validate a formal specification is shown in Figure 4.2. The developer is given an informal specification and once he is aware of the outline of his conception, he formulates the skeleton of his specification in the formal language. This formal specification can be validated and simulated using tools. The formal specification is later used as a model for the implementation [94, 95].

![Diagram](image)

**Fig. 4.2 Software Development using Formal Description Techniques [93]**

The flowcharts and structural diagrams that are used in conventional software development are basically a bit like formal specification. The difference with the FDT’s is that there are tools that can be applied to a specification.

### 4.3 PROTOCOL SPECIFICATION WITH PROMELA

PROMELA provides a suitable specification language for concurrent systems. Process interaction and process coordination are at the very basis of the language. It has three basic types of objects:

- a. Processes: They specify the behavior.
- b. Variables: They define the environment in which the processes run.
- c. Channel: Message channels are used for modeling the inter-process communication.
The communication between processes takes place via either message or shared variable. A protocol in PROMELA is defined by a set of processes. A process consists of statements on a set of local and global variables. The PROMELA also supports sending and receiving of messages via communication queries. The brief discussion of the language is as under:

4.3.1 Basic Object Types
The syntax of PROMELA is simple and compact but nevertheless surprisingly powerful in its expressive capability. The following three types of objects are used to construct a PROMELA specification:

a. **Processes** use an extended finite state machine model for the description of its behavior. The concepts of the processes are very close to Communicating Sequential Processes (CSP) [97, 98]. All processes are on the same level. Processes are described by the “proctype” construct, which introduces a process prototype. A special “init” process is always invoked by the system at start-up. Additional processes can be dynamically invoked during execution using the “run” statement. The same process prototype can be invoked multiple times; this makes it possible to build recursive models.

b. **Variables** can be defined either globally or locally. By defining them globally, they can be used to interchange data between the different proctypes. This can be used to describe a communication via shared memory. A variable definition contains the type of the variable. It is possible to declare arrays of variables. However, there are no structured variables like the “struct” in C.

c. **Channels** are an exceptional data type. They are essentially finite-length FIFO queues. Like variables, channels can be defined globally for the specification or locally within each proctype. The channels can transmit messages of fixed, predefined types. Channels allow a synchronization of two processes. The channels should be used as the main medium for the modelization of all communication in the specification.
4.3.2 Processes

```c
proctype ProcessA(int x, bool flag)
{
    /* Body of Process A’s definition */
}

proctype ProcessB(byte y)
{
    /* Body of Process B’s definition */
}

init
{
    run(ProcessA(5, true));
    run(ProcessB(3));
    printf(“Init done.\n”)
}
```

Fig. 4.3 Skeleton of a PROMELA Program

Figure 4.3 shows a skeleton of a typical PROMELA program. This program consists of two (empty) process prototype definitions. The “init” process is invoked when the program is started and instantiates the two processes, “Process A” and “Process B” by executing the “run” statements. Afterwards it writes a message on standard output. As one can see in the example, process prototype definitions are just descriptions of the structure of a process. In order to actually use those prototypes the “run” statement is necessary.

4.3.3 Variables and Types

As mentioned, variables can be defined either globally or locally for the process prototypes. All definitions that are not inside of a “proctype” or “init” construct are global and can be accessed from all processes. A PROMELA variable is one of the following five predefined types, that is: bit, byte, short, int or chan.

In Figure 4.4, some examples for variable definitions are given. Here, the first line declares a variable “a” which is initialized with the value 1. The second line serves to declare an array of 10 integers. The third declares the variable b as byte and does not initialize its
value. The following two lines are an example for the definition of a channel and are explained in the next paragraph.

```
int a = 1;
int b[10];
byte b;

mtype = {ack, nak, cr, dr, conn, disc}
chan canal = [5] of {chan, mtype}
```

**Fig. 4.4 Example for Variable and Channel Declarations**

### 4.3.4 Channels

A channel declaration defines data types of the messages that can be transmitted over a channel. The data types may contain channel names; it therefore can be very useful for the specification of recursive models. Channels may either be defined as synchronous ("rendezvous") or asynchronous. Synchronous channels are declared by setting the length of the channels FIFO queue as zero and can be used to synchronize two processes. A write access to a synchronous channel blocks the writing process until a corresponding read access is executed by another process or vice versa.

### 4.3.5 Control Structures

It is very easy to specify distributed automata in PROMELA that can communicate using either message channels or shared memory. Due to closeness of the language to automata theory properties of finite automata’s can easily be translated into PROMELA specifications.

### 4.3.6 Executability

One key concept of the PROMELA language is the *executability* of statements. In PROMELA, there is no difference between conditions and instructions. If a condition is true, it is executable. Equally an instruction can or cannot be executable. For example, a receive operation on an empty channel is not executable. It is not necessary although that a non-executable transition will never be executed. It might become executable later on under conditions such as ‘if content of the channel changes’.
In fact, a condition as well as a statement has to be seen as the analogue of a transition in the finite state machine model. If a transition between two states is executable, the state machine can move from one state to the other one. If such a transition changes the state variables (which might also be FIFO queues) of the machine, one would have to call it an instruction. If no variables etc. are changed, it could be called an expression. As a result, it is easy to translate linear sequences of transitions into a computer language. In PROMELA, statements and expressions are separated by “;” (the semicolon) or “->”. Both of these constructs have exactly the same meaning, which is that the statements that are separated are to be executed sequentially.

4.3.7 Non-Determinism

For the validation of a specification, the most important aspect is that nothing must be considered as impossible. If it is possible for an event to occur, it always has to be considered that it actually may occur, even if the probability is very small. Figure 4.5 shows a state machine with a non-deterministic choice, for example, for value of the variable a as 1, all three expressions are “true” and therefore all of them are executable.

![Fig. 4.5 Non-Deterministic Possibilities in a Finite State Machine](image-url)
4.4 VISUAL INTERFACE TO PROMELA - VIP

The VIP tool [99] was designed by Moataz Kamel and Stefan Leue, in consultation with Gerard Holzmann. This tool was implemented by Moataz Kamel. This tool is a Java based graphical front end to the PROMELA specification language and the SPIN model checker. It makes use of two components: A visual editing component called Nexus that was jointly developed by Moataz Kamel and Chris Trudeau, as well as an XML-based serialization library called xml4j designed and implemented by Michael Abd-El-Malek. VIP can be run under any operating system environment for which an implementation of the Java Runtime Environment is available.

All VIP models have the extension .xml and actually the models are being stored in an XML format representing the internal object structure. When defining the structure of the VIP model it will be necessary to connect capsule instances via ports and these ports have to be defined to have a type, namely that of one of the protocol classes. A VIP model consists of three major types of classes:

- Capsule classes, which define structure and behavior of the concurrent objects within a VIP model, called capsules.
- Protocol classes, which define signal lists to be exchanged amongst capsules.
- Data classes, which permit the definition of some basic data types. The properties of a data class are either one of the basic Promela types, or a typedef.

VIP supports a visual formalism called v-PROMELA which extends the PROMELA language with a graphical notation to describe structural and behavioral aspects of a system. It also introduces hierarchical modeling and object-oriented concepts. The v-Promela language borrows many concepts from UML RT, proposed by Jim Rumbaugh and Bran Selic. The structural part of v-PROMELA model consists of structural elements called capsules and describes their interconnection and hierarchical nesting using a variant of UML collaboration diagrams. The behavioral aspects of a v-PROMELA model are described by hierarchical communicating extended finite state machines and support such features as group transitions and optional return to history from group transitions.
The VIP tool provides a graphical v-PROMELA editor that supports point and click editing of v-PROMELA structure diagrams and hierarchically nested state machines. The editor incorporates syntax checking to warn the user about in-correct use of v-PROMELA graphical syntax.

4.5 ASSUMPTIONS

The following assumptions about wireless sensor nodes and WSN have been made while specifying the model of the architecture:

a. The network is deployed in a two-dimensional field of finite area. This assumption is mostly for the convenience of discussion and does not prevent the analysis from being applied to a higher dimension.
b. Nodes are capable of measuring the signal strength of a received message [100].
c. Each node knows its location.
d. Nodes consume energy when transmitting, receiving and sensing, but not while idling.
e. A unique node is designated as a cluster-head of each cluster. Every node belongs to a cluster and no node belongs to multiple clusters.
f. Communication links are bi-directional and unreliable.
g. Underlying protocols ensure that nodes are aware of the set of nodes within their transmission radius.
h. Sensors can have a controlled variable sampling rate.

4.6 MODEL OF SENSOR NODE

In the sensor networks, sensors can cooperate with each other to conduct computational tasks based on the input they supply to each other and/or values supplied to the sensors by their environment. It is reasonable to assume that both communication links and sensors themselves are unreliable. Different actions of sensor nodes executes at specific timestamp when the sensor nodes are in active state. It periodically sends the reading to its cluster head. Upon receiving a message, the sensor node updates its local variables and may take on further actions like go into the sleep state or may change the periodicity of sensor readings. Other actions of the sensor node can be advertising itself as new cluster head and some processing actions.
4.6.1 Data objects of Sensor Node (Working as CH)

These are listed as under:-

1. Sensor Node Location
2. Energy of the node at the beginning
3. Energy usage for packet transmission
4. Energy usage for packet receipt
5. Node Status - Active, Sleeping
6. Probing Range
7. ARS Location - When a node gets elected as a CH, it probes the environment to determine location of the nearest ARS/parent CH and gets connected to it
8. Cluster Head Alive Timer - After expiry of this timer, the sensor node working as a leader ceases to be CH and starts leader election algorithm.
9. BatchTimer
   • This timer is relevant to the sensor node if it is currently the leader of its cluster.
   • The CH reads all the records from its buffer after the expiry of this timer.
   • It aggregates the sensed information location wise.
   • After aggregating the information, it sends it to the destination.
10. ProbeARSTimer
    • The CH sends the probe message for the ARS to its environment.
    • After expiry of this timer, the cluster head assumes that there are no active ARS nodes.
11. EnergyLeftProbeTimer
    • Periodically, the cluster head checks amount of energy left.
    • It ceases to be cluster head if its battery level falls below a threshold limit.

Working of Cluster Head

The working of cluster head is shown in Figure 4.6. It periodically receives sensed information from member sensor nodes according to its TDMA schedule. A new sensor node sends Join request to it whenever it comes near its vicinity. On receiving the request it
changes the TDMA schedule accordingly so that it may collect data from the new sensor node also.

Periodically CHs should be rotated after few rounds of operations. Current CHs collects candidature request from sensor nodes reporting to it and runs leader election algorithm. The leader election algorithm chooses best candidate as a new CH. After its election, new CH broadcast his selection information to all nearby sensor nodes and sends join requests to them. In case of hierarchical clustering information is also sent to child CHs. Sensor nodes, which accepts its invitation confirms their joining by sending a message to it.

The clustering process for Centralized and Hierarchical clustering as used in this work is explained in Chapter 6 on Simulation and Result Analysis.

Fig. 4.6 Working of Cluster Head

4.6.2 Various possible States of Sensor Node working as ordinary node

a) Probing: The sensor node starts probing its neighbourhood whenever it wakes up.
b) Sleeping: The sensor node periodically goes for sleep to conserve energy.
c) Active: In this state, sensor node periodically probes environment.
4.6.3 Data Objects of Normal Sensor Node

These are described as below:-

1. Sensor Node Location
2. Current battery level of the node
3. Energy usage for packet transmission
4. Energy usage for packet receipt
5. Node Status - Active, Sleeping
6. Probing Range
7. Cluster Head Location
8. Sensor Reading Timer - The frequency of the probation of environment for reading depends upon this timer.
9. Active Timer - It specifies the time for which the sensor node will probe environment for data. After expiry of this timer sensor node goes to sleeping mode again.
10. Wakeup Timer - It determines amount of time the sensor node will sleep. On the expiry of this timer, the sensor node starts working again
11. Advertise Period Timer - It determines the amount of time an un-clustered sensor node or node waits before it sends Join Message to some existing CH.
12. Join Period Timer - It determines the amount of time for which current sensor node has to try joining some existing CH. After expiry of this timer the current node will become the CH itself and sends join message to other sensor nodes.

Working of Sensor Node

A sensor node spends its time between sleep and active states. While in active state the sensor node senses and collects data at regular intervals from its environment. The sensed reading/data is then transferred to the CH for further necessary processing. The sensor node will change various timer values upon receiving appropriate instruction from its CH. The detailed working of normal sensor node is shown in Figure 4.7.
4.6.4 Performance of Sensor Node

The parameters probing range and wakeup rate controls performance of the sensor node as explained below:

a. The desired redundancy can be achieved by setting the corresponding probing range. For example, an application requiring high robustness may choose a small probing range to achieve high density of working nodes.

b. The number of wakeups decides the overhead, thus it should be kept low. However, if the wakeup rate is set to be too low, when a working node fails unexpectedly, there can be large gaps during which no working node is available.

c. Each probing node adjusts its wakeup rate according to the observation of its sleeping neighbours, so that transient node failures are tolerated by the application.

4.7 MODEL OF CLUSTER

In a large-scale network, if all the nodes have to communicate their data to their respective destination, it will deplete their energy quickly due to the long-distance and multi-hop nature of the communication. This will also lead to network contention. The clustering is a standard approach for achieving efficient and scalable control in these networks.
Clustering results in number of benefits. It facilitates distribution of control over the network. It saves energy and reduces network contention by enabling locality of communication: nodes communicate their data over shorter distances to their respective cluster-heads. The CHs aggregate these data into a smaller set of meaningful information. Not all nodes, but only the CHs need to communicate far distances to their respective ARS. In the proposed architecture:

a. The sensor Nodes forms clusters periodically
b. Whenever the node wakes-up it sends a probe message to join existing cluster.
c. The sensor nodes within the cluster only communicate with their respective cluster head
d. The cluster head responsibility is rotated among the sensor nodes to increase the longevity of the sensor node

4.7.1 Re-clustering Process

The CH position is periodically rotated among the various sensors to prevent draining of battery of a single sensor. Periodically the CH checks the energy level of self and its member nodes. It submits the candidatures of self and its member nodes for selection as CH in next round. Moreover, whenever a new node joins WSN it probes its neighbourhood for new cluster with a CH. As a result it either joins the existing cluster or goes into the sleeping state to conserve its energy. It again tries after waking up until it is able to join some CH. When a sensor node is selected as a CH it sends join request to all the nodes in its vicinity. The nodes that send join confirmation become its members. The new CH prepares a TDMA schedule according to the list of its members. The detailed re-clustering process is shown in Figure 4.8.

4.7.2 Data Objects of Cluster Class

As shown in Figure 4.9, a cluster consists of one cluster head and multiple sensor nodes. The Cluster class is a logical entity. Every CH maintains the IDs of sensor nodes reporting to it and temporarily stores aggregated and summarized reading.
Fig. 4.8 Re-Clustering
Fig. 4.9 Structure of Cluster Class
4.8 MODEL OF ARS

Every ARS supports two types of interfaces – ad-hoc relay interface and cellular interface. Using ad-hoc interface, ARS communicates with other ARSs and sink nodes. It uses cellular interface to communicate with base stations of cellular network. The “Data Writer” component of ARS passes the information received from various CHs of WSN to the corresponding base stations. It may also store the information received in its local buffer and after some time send the summarized information to BS to further conserve power and communication bandwidth.

The effective action taken by ARS depends upon the amount of data it is able to obtain pertaining to the critical area. In order to minimize the risk of losing sensitive data, same data is replicated in adjacent BSs; so that if a BS collapses due to any reason the ARS can obtain the required information from elsewhere. The finite state machine of ARS is explained in Figure 4.10.

![Finite State Machine of ARS]

Fig. 4.10 Finite State Machine of ARS
4.8.1 Working of ARS

Each ARS maintains a list of CH reporting to it and ARSs within its vicinity. Each CH periodically sends sensed data to ARS. The ARS sends received information to BS of each adjacent cell to which it is connected. It may also take automatic action in case of emergency on the basis of stored information. It determines the current situation as emergency when average reading value overshoots a threshold value in a specific area for a specified period of time.

In case of emergency it also requests its neighbouring ARSs for data so that it has comprehensive knowledge of the critical area in which action has to be taken. On receiving the data about the critical area it takes appropriate action for a specified duration of time. Periodically it probes the BSs to determine whether or not they are still alive. In case it determines that no BS is alive it sends the relinquish control message to all the CHs reporting to it so that they can join some other ARS. When ultimately it finds that it can join some BS it sends the rejoin message to the CHs in its vicinity. The detailed working of ARS has been shown in Figure 4.11.

4.8.2 Data Objects of ARS Class

The various data objects of ARS class are listed as under:-

1. Location of ARS Node
2. Status of ARS
3. Action Timer
4. Radius of coverage area in which action has to be taken.
5. Energy Left
6. Status of Base stations on each side of the edge
7. ProbeBSCompletion timer after expiry of which ARS assumes that corresponding base station has collapsed.
8. ProbeBSStart Timer controls the periodicity of the probe.
Fig. 4.11 Working of ARS
4.9 MODEL OF ACTION UNIT CLASS

As shown in Figure 4.12, the Action Unit class consists of group of Clusters and ARS. All the clusters belonging to the Action Unit sends data directly to the ARS. The ARS then forwards the information received to the Base Station.

4.9.1 Data Objects of Action Unit Class

1. The list of CH Locations
2. Location Id of adjacent BSs
3. List of Neighbouring ARSs within the probe area

4.10 MODEL OF BASE STATION

The issues of speedier storage and retrieval as well as enhanced availability are extremely important for taking effective decision in a real world mission critical WSN
based applications. The role of BS therefore is very important in the proposed model. The finite state machine for the BS is illustrated in Figure 4.13.

![Finite State Machine for Base Station](image)

**Fig. 4.13 Finite State Machine for Base Station**

### 4.10.1 Working of Base Station

One of its main responsibilities is to store efficiently the information received from various ARSs reporting to it. It uses two methods to reduce the amount of information that has to be maintained by it – by purging the old records and storing aggregate of the information received over a period of time area wise.

The ARSs in crisis request the BS for information so that effective and comprehensive action can be taken in critical areas. Providing timely information to the ARS is therefore another major responsibility of BS.

It periodically checks the status as well as performance of the ARSs connected to it and may recommend update of their control parameters to enhance their effectiveness.
Fig. 4.14 Working of Base Station
Every BS in the WSN is connected to the sink. In case it fails to detect the sink it send relinquish control message to all the ARSs connected to it. It then periodically probes for the sink. When ultimately it detects the sink it again sends the rejoin message to the ARSs. The detailed working of Base-Station is shown in Figure 4.14.

4.10.2 Data Objects of Base Station Class

1. Location of Base Station
2. The starting and ending coordinates of the rectangle enclosing the cell of the base station.
3. Database of records.
4. Storage Timer, after the expiry of which, it reads all the records pertaining to a location area, aggregates them and then stores the aggregated information.
5. Purge timer, on the firing of which, it deletes old records.
6. Retention Period for the records.

![Fig. 4.15 Structure of Cell]
4.11 MODEL OF CELL

As shown in Figure 4.15, a cell has one base station and six Action Units. Again like cluster class, the cell class is also a logical entity.

4.11.1 Data Objects of Cell class

1. Location of Base Station
2. List of locations of ARS nodes.

4.12 MODEL OF SINK

The sink supervises and synchronizes the working of various components of WSN. The finite state machine for sink is shown in Figure 4.16

Fig. 4.16 Working of Sink

4.12.1 Working of Sink

1. It maintains a list of all BSs and their current status.
2. It maintains a list of all ARSs and their current status
3. It maintains a list of all clusters and their current status.
4. It stores statistical information to control the operation of WSN.
5. It periodically obtains the status of all the components of WSN.
6. On the basis of statistical information, it changes the value of performance parameters to improve the behavior of WSN.
7. From time to time, it provides the current status of WSN to the controlling authority.
8. It requests the controlling authority to replace faulty components of WSN.

4.12.2 Data Objects of Sink class
1. Location Identifier
2. Statistical Information
3. EnquiryStatusTimer for initiating probe of WSN.

4.13 MODEL OF REAL TIME WSN (RTWSN)
This class represents entire WSN. It has one sink and many cells. The structure of RTWSN is depicted in Figure 4.17 as under.

![Fig. 4.17 Structure of RTWSN](image)

4.13.1 Data Objects of RTWSN class
1. Location of Sink
2. List of locations of all Cells.
3. List of all parameters
4.14 FORMAL VERIFICATION USING SPIN

Formal verification is the process of checking whether a design satisfies desired requirements [101, 102]. It conducts an exhaustive exploration of all possible behaviors of the system. Formal verification works on models (rather than implementations). It is supported by tools [103]. SPIN model-checker is one of the tools which can be used for formal verification. The jSpin [104] GUI interface for SPIN model checker has been used, for verifying the model of the architecture, because of its powerful model checking capabilities. All the aspects of jSpin are configurable: some at compile time, some at initialization through a configuration file and some at runtime.

In this thesis we have modeled the proposed architecture using PROMELA and used jSpin for model checking. The model consists of around 1650 lines of PROMELA code excluding comments. The output of jSpin proved that the model of proposed architecture works according to its intended behavior. Since the verification is bounded by the amount of physical memory available, complexity of the system was reduced. The modular approach allowed separate verifications for each component of model and of composite model, in order to prove properties for the complete model.

4.15 CONCLUSION

Model checking is a fully automated technique for the verification of finite state systems and to verify the correctness of software designs. The working of sensor networks can be described as a set of concurrent communicating processes in PROMELA. The SPIN is a model checking method and appropriate tool for verifying the sensor networks specifications. In this chapter formally the architecture of reliable and real time data placement model has been described using a GUI based tool VIP. The detailed finite state machines of various components have been described. The role each component plays in the working of overall system has been explained.
5.1 INTRODUCTION

This chapter describes statistical/mathematical estimates of parameters that affect working of various components of the proposed model. The issue of maintaining a similar time among various components of WSN is explained in this chapter. The sensing, communication and coverage of sensor nodes is also expressed. The issues of average number of participating sensors required at a time and number of ARS needed to cover the entire system is also explored. The basic clustering parameters, as well as cluster head rotation timeout and optimum number of clusters per ARS are also explained. Further, energy dissipation of sensor node in various activities is analyzed.

5.1.1 Formulation Notation

The notations used in different equations specified in this chapter, are as under:

- \( N \): index set of sensor nodes,
- \( r_i \): broadcast range of sensor node \( i \in N \),
- \( d_{ij} \): distance from sensor node \( i \in N \) to sensor node \( j \in N \),
- \( f_i \): distance from sensor node or CH \( i \in N \) to the ARS,
- \( b_i \): battery level of sensor node \( i \in N \) (J),
- \( l \): data packet size (bits),
- \( E \): energy spent by transmitting and receiving circuitry (J/bit),
- \( E_{DA} \): coefficient for data aggregation (J/bit/signal),
- \( n \): number of sensor nodes having a positive battery level,
- \( k \): number of clusters
- \( \alpha \): parameter to determine CH candidates (\( 0 < \alpha \leq 1 \)),
- \( M^2 \): wireless sensor network region
5.1.2 Decision Variables

The following decision variables are used in this chapter, in various equations that describe working of various components of the proposed model.

\[ \text{for } i, j \in N \]

- \( x_i \): binary variable such that \( x_i = 1 \) if sensor node \( i \) is selected as a CH and \( x_i = 0 \) otherwise.
- \( y_{ij} \): binary variable such that \( y_{ij} = 1 \) if sensor node \( i \) belongs to the cluster where sensor node \( j \) is a CH and \( y_{ij} = 0 \) otherwise.
- \( c_{ij} \): binary variable such that \( c_{ij} = 0 \) if sensor node \( i \) belongs to the cell \( j \) otherwise \( c_{ij} = 1 \).
- \( a_{ij} \): binary variable such that \( a_{ij} = 0 \) if sensor node \( i \) is currently attached to the ARS \( j \) and \( a_{ij} = 1 \) otherwise.
- \( S_i \): 0 if sensor node \( i \) has a positive battery level and 1 otherwise.
- \( w_{ij} = \begin{cases} 0 & \text{sensor } j \text{ is asleep at time } i \\ 1 & \text{sensor } j \text{ is awake at time } i \end{cases} \)

5.1.3 Amplifier Energy

Amplifier Energy (\text{pJ/bit/mn}) can be defined as the energy required by transmitter amplifier to maintain an acceptable signal-to-noise ratio in order to transfer data messages reliably. The transmitter amplifier has to spend this energy to overwhelm radio propagation loss.

Depending on the distance between the sensor nodes, either free space or two ray ground propagation model can be considered to approximate the path loss sustained due to wireless channel transmission. As in [59], given a threshold transmission distance of \( d_0 \) free space model will be employed for \( d < d_0 \) and multipath fading model will be used for \( d \geq d_0 \). As used in [59], for both models, amplifier energy required by the transmitter amplifier for data transmission from sensor node \( i \) to \( j \) (\( D_{ij} \)), from sensor node \( i \) to the ARS (\( F_i \)) and to broadcast a packet to its range (\( R_i \)) can be given by the following equations:
\[ D_{ij} = \begin{cases} \varepsilon_{FS}d_{ij}^2 & \text{(if } d_{ij} < d_0) \\ \varepsilon_{MP}d_{ij}^4 & \text{(if } d_{ij} \geq d_0) \end{cases} \quad (5.1) \]

\[ F_i = \begin{cases} \varepsilon_{FS}f_i^2 & \text{(if } f_i < d_0) \\ \varepsilon_{MP}f_i^4 & \text{(if } f_i \geq d_0) \end{cases} \quad (5.2) \]

\[ R_i = \begin{cases} \varepsilon_{FS}r_i^2 & \text{(if } r_i < d_0) \\ \varepsilon_{MP}r_i^4 & \text{(if } r_i \geq d_0) \end{cases} \quad (5.3) \]

where \( \varepsilon_{FS} \) and \( \varepsilon_{MP} \) denotes transmit amplifier parameters corresponding to free-space and multipath fading models respectively. The meaning of notations \( d_{ij} \), \( f_i \) and \( r_i \) is already explained in section 5.1.1. As per [59], \( d_0 \) is threshold distance that can be determined as under:

\[ d_0 = \sqrt{\varepsilon_{FS}/\varepsilon_{MP}} \quad (5.4) \]

The sensor node will consume \( lE \) energy for receiving or transmitting \( l \) bit size packet data where \( E \) is the energy spent by the transmitting or receiving circuitry. The total amount of energy consumed \( E_{ij} \) in transmission of \( l \) bit size packet data from sensor node \( i \) to \( j \) therefore will be

\[ E_{ij} = lE + D_{ij} \quad (5.5) \]

**5.1.4 Clustering**

In order to conserve energy, group of sensor nodes forms a cluster. The cluster-head position is also periodically rotated among sensor nodes to prevent draining of the battery of a single sensor. In the model, the ARS organizes the network. The operation of cluster based sensor network is divided into rounds. Each round has two phases – clustering phase and data transmission phase. Further, the rounds are repeated to monitor events continuously.

**5.2 TIME MEASUREMENT**

An internal clock reference is used to measure real time \( t \) from a local oscillator that is equipped with sensor nodes. As depicted in Figure 5.1, due to various physical effects, an oscillator is subjected to frequency drifts. As a result, only an estimation \( E(t) \) of the real time \( t \) can be obtained.
As specified in [106], even if frequency of an oscillator changes over time; real time estimation can be approximated with good accuracy using equation 5.6, if time intervals under measurement are small.

\[ E(t) = (1 + \delta)t + \mu \]  

(5.6)

where \( \delta \) is the clock drift relative to the correct rate and \( \mu \) is the clock offset. The rate of a perfect clock, \( \frac{dE(t)}{dt} \), would be equal to 1 when \( \delta = 0 \). The clock performance expressed in terms of part per million (ppm), is defined in [106] as the maximum number of extra or missed clock counts over a total of \( 10^6 \) counts, that is, \( \delta \times 10^6 \).

**Fig. 5.1 Relationship between the Measured time and Actual time [106]**

Suppose a node generates a time delay of \( \tau_d \) seconds, effective generated delay \( \tau_{\text{eff}} \) in the presence of a clock drift \( \delta \), then as in [106] will be as under:

\[ \tau_{\text{eff}} = \frac{\tau_d}{1+\delta} \]  

(5.7)

Verdone et al. [106] has given equation 5.8 to specify estimated value \( E(\tau) \), whenever a node has to measure a time estimation for duration \( \tau = t_2 - t_1 \) seconds:

\[ E(\tau) = E(t_2) - E(t_1) = \tau(1 + \delta) \]  

(5.8)

In both cases there is no dependence on the clock offset \( \mu \).
5.2.1 Time Synchronization

Like in any distributed computer system, time synchronization is a critical issue in sensor networks also. It is necessary to maintain a similar time with a certain tolerance among various components throughout the lifetime of the network. One of the most popular techniques used to maintain time synchronization is Timing-sync Protocol for Sensor Networks (TPSN) [107]. TPSN works in two phases – the first phase is level discovery and the second phase is synchronization. The aim of the first phase is to create a hierarchical topology in the network, where each component is assigned a level. Only one component, the root is assigned level 0. In the second phase, a component at level $i$ synchronizes to its parent at level $i - 1$ by exchanging time-stamp messages via two-way handshake approach as depicted in Figure 5.2.

\[ \delta = (g_4 - g_1) - (g_3 - g_2) \]  
\[ \theta = \frac{(g_2 - g_1) + (g_3 - g_4)}{2} \]

Fig. 5.2 Two-way message handshake. [107]

At the end of handshake time $g_4$, child component A obtains times $g_1, g_2$ and $g_3$ from the acknowledgement packet. The times $g_2$ and $g_3$ are obtained from the clock of parent component B whereas times $g_1$ and $g_4$ are generated by component A itself. As suggested by [108], measured round trip delay $\delta$ and clock offset $\theta$ of A relative to B after processing of the acknowledgment packet, can be given by equation 5.9 and 5.10 respectively as under:
The root node sends a time sync packet to initialize the time synchronization process. The synchronization process continues until all other components are synchronized to root and network-wide synchronization is achieved.

5.3 SENSING, COMMUNICATION AND COVERAGE

In this section, the basic mathematical framework for sensing, communication and coverage of sensor nodes applicable to the designed model have been explained.

5.3.1 Sensing

Each node has a sensing gradient, whose radius, although ideally extending to infinity, attenuates gradually as the distance increases. [109] has given a method specified in equation 5.11 to determine sensitivity $S$ of a sensor $s_i$ at point $P$.

$$S(s_i, P) = \frac{\lambda}{[d(s_i, P)]^\gamma}$$  \hspace{1cm} (5.11)

where $\lambda$ and $K$ are positive sensor-dependent parameters and $d(s_i, P)$ is the Euclidean distance between the sensor and the point. Typically the value of $\gamma$ is dependent on environmental parameters and varies between 2 and 5. Since the sensitivity rapidly decreases as the distance increases, a maximum sensing range for each sensor can be defined.

![Diagram](image)

Fig. 5.3 (a) Probabilistic sensing model; (b) communication model. [110]

A binary sensing model is assumed, according to which a sensor is able to sense from all points that lie within its sensing range. The sensing range for each sensor node is
therefore confined within a circular disk of radius $R_s$. In a heterogeneous sensor network, the sensing radii of different types of sensors might vary, but to simplify the analysis of coverage, it is assumed that all nodes are homogeneous and maximum sensing radius for all of them is $R_s$.

As suggested in [110], binary sensing model can be extended to more realistic probabilistic sensing model illustrated in Figure 5.3 (a). Let us define a quantity $R_u < R_s$, such that the probability that a sensor would detect an object at a distance less than or equal to $R_s - R_u$ is 1 and at a distance greater than or equal to $R_s + R_u$ is 0. In the interval $R_s - R_u, R_s + R_u$, there is a certain probability $p$, that an object will be detected by the sensor. The quantity $R_u$ is a measure of uncertainty in sensor detection. This probabilistic sensing model reflects the sensing behavior of devices such as infrared and ultrasound sensors.

### 5.3.2 Communication

Similar to the sensing radius $R_s$, $R_{cl}$ is a communication radius as depicted in Figure 5.3 (b) for each sensor $s_i$. [110] specified that two sensors, $s_i$ and $s_j$, will be able to communicate, only if Euclidean distance between them is less than or equal to the minimum of their communication radii, that is, when

$$d(s_i, s_j) \leq \min\{R_{ci}, R_{cj}\}. \quad (5.12)$$

This basically means that the sensor with smaller communication radius falls within the communication radius of other sensors. Nodes that are able to communicate with each other are called one-hop neighbours. The communication radii might vary depending on residual battery level of an individual sensor. Let $R_c$ be the communication radii for all the nodes.

### 5.3.3 Coverage Model

Depending on the sensing range, an individual node will be able to sense a part of the sensing field only. The probabilistic coverage of a point $P(x_i, y_i)$ by a sensor $s_i$ has been defined by [110] using following equation:
\[
C_{x,y_l} = \begin{cases} 
0, & R_s + R_u \leq d(s_l, P) \\
\exp^{-\gamma a^\beta}, & R_s - R_u \leq d(s_l, P) < R_s + R_u \\
1, & R_s + R_u \geq d(s_l, P)
\end{cases}
\quad (5.13)
\]

Here, \(a = d(s_l, P) - (R_s - R_u)\) and \(\gamma\) and \(\beta\) are parameters that measure detection probabilities when an object is within a certain distance from the sensor. All points that lie within a distance of \((R_s - R_u)\) from the sensor are said to be 1-covered and all points lying within the interval \(R_s - R_u, R_s + R_u\) have a coverage value less than 1 that exponentially decreases as the distance increases as per equation 5.13. Beyond the distance \(R_s + R_u\), all the points have 0-coverage by this sensor. However, a point might be covered by multiple sensors at the same time, each contributing a certain value of coverage. Hence, total coverage of a point has to be determined.

**Total Coverage of a Point**

Let \(S = \{s_i, i = 1,2,\ldots,k\}\) be the set of nodes whose sensing ranges cover the point \(P(x_l, y_l)\). As in [110], total coverage of point \(P\) can be determined as:

\[
C_{x,y_l}(S) = 1 - \prod_{i=1}^{k} \left(1 - C_{x,y_l}(S_i)\right)
\quad (5.14)
\]

Since \(C_{x,y_l}(S_i)\) is the probabilistic coverage of a point as defined in Equation 5.13, the term \(\left(1 - C_{x,y_l}(S_i)\right)\) is the probability that the point is not covered by sensor \(s_i\). Now, since probabilistic coverage of a point by one node is independent of another node, the product \(\prod_{i=1}^{k} \left(1 - C_{x,y_l}(S_i)\right)\) of all such \(k\) terms will denote joint probability that the point is not covered by any of the nodes. Hence, one minus this product would give probability that point \(P\) is covered jointly by its neighbouring sensors and is defined as its total coverage. The total coverage of a point lies in the interval \([0, 1]\).

**5.4 AVERAGE NUMBER OF PARTICIPATING SENSORS**

Let \(A_i\) be a random variable that denotes the number of awake sensors and \(E[A_i]\) be the average number of awake sensors at time \(i\). Let sensors sleep for a random amount of time from \(s\) to \(S\) and stay awake for another random amount of time from \(l\) to \(L\). It is assumed that the sensor node sleeps and wakes up alternately.
If probability that sensor $j$ is awake at time $i = P_i$, then from [105]

$$w_{ij} \sim Bernoulli \ (P_i) \Rightarrow E[w_{ij}] = P_i$$

(5.15)

where $w_{ij} = \begin{cases} 0 & \text{sensor } j \text{ is asleep at time } i \\ 1 & \text{sensor } j \text{ is awake at time } i \end{cases}$

Since sensors sleep and wake up independently of each other, from [105],

$$A_i \sim Binomial \ (n, P_i) \Rightarrow E[A_i] = n \cdot P_i$$

(5.16)

To evaluate $P_i$, it is assumed that a sensor during its lifetime goes through $m$ cycles of different lengths. In each cycle, a sensor sleeps for a random amount of time from $s$ to $S$ and stays awake for another random amount of time from $l$ to $L$. If $m$; total number of cycles; is large enough, then the total awake time, the total sleeping time and the total life time of a sensor can be expressed as specified in [105] :-

Awake Time $= \frac{m \cdot (l+L)}{2}$

(5.17)

Sleeping Time $= \frac{m \cdot (s+S)}{2}$

(5.18)

Life Time $= \frac{m \cdot (l+L)}{2} + \frac{m \cdot (s+S)}{2} = \frac{m \cdot (l+L+s+S)}{2}$

(5.19)

As in [105], using equations 5.17 and 5.19, the probability $P_i$ that a sensor is awake at time $i$ can be given by equation 5.20:

$$P_i = \frac{\frac{m \cdot (l+L)}{2}}{\frac{m \cdot (l+L+s+S)}{2}} = \frac{l+L}{l+L+s+S}$$

(5.20)

Similarly, using equations 5.18 and 5.19 the probability $1 - P_i$ that a sensor is asleep at time $i$ can be given by equation 5.21:

$$1 - P_i = \frac{\frac{m \cdot (s+S)}{2}}{\frac{m \cdot (l+L+s+S)}{2}} = \frac{s+S}{l+L+s+S}$$

(5.21)

It is evident from equations 5.20 and 5.21 that the probability a sensor is awake or asleep at time $i$ is independent of $i$, therefore $P$ can be used instead of $P_i$. As in [105], average number of awake sensors at time $i$, can be specified as under:

$$E[A_i] = \frac{n \cdot (l+L)}{l+L+s+S}$$

(5.22)
However, average metric alone is not enough; how far is this average from the actual number of awake sensors at that time also need to be determined. As in [105], this can be determined by taking variance of $A_i$ using elementary statistics as given in equation 5.23 below:

$$\sigma^2[A_i] = N \cdot P \cdot (1 - P) \quad \text{as } A_i \sim Binomial(n, P)$$  \hspace{1cm} (5.23)

### 5.5 NUMBER OF ARSs NEEDED TO COVER THE ENTIRE SYSTEM

The issue of number of ARSs needed to cover the entire system is discussed in [111]. The ARS is placed on each pair of shared edges along the border between two cells. Since an ARS is shared by two cells and each cell has 6 edges, it is obvious that at most $3N$ ARSs are needed in an $N$-cell system. As specified in [111], total number of ARSs needed for an $N$-cell system can be given by equation 5.24.

$$3N - \lfloor 4\sqrt{N} - 4 \rfloor$$  \hspace{1cm} (5.24)

Cells can be classified in two categories - boundary cell and non-boundary cell. The boundary cell has some non-shared edges. In the proposed architecture, ARS is placed on every edge without considering whether it is a shared edge or non shared edge.

### 5.6 ANALYSIS OF ENERGY DISSIPATION DURING CLUSTERING

In the clustering process four types of energies are dissipated. They are:

1. $E_{CH}$: Total energy spent by a CH in a given data transmission round for useful work.
2. $E_{mem}$: Total energy spent by a cluster member node in a given data transmission round for useful work.
3. $E_{CH-ooh}$: Energy overhead that a CH node has to spend in each cluster setup phase.
4. $E_{mem-ooh}$: Energy overhead that a non CH node has to spend in each cluster setup phase.

As specified in [63], various energies spent by a $CH_i$ in a given data transmission round for useful work can be determined as under:

- Energy $E_{CH-Rx}$ spent in receiving signals from member nodes
Data Aggregation Cost $E_{CH-Da}$

$E_{CH-Da} = l \times E_{Da} \times \left( \frac{n}{k} \right)$ (5.26)

- Energy $E_{CH-Tx}$ spent in transmission of Aggregated signal to ARS

$E_{CH-Tx} = l \times E + l \times F_i$ (5.27)

Equations 5.25 to 5.27 use notations specified in section 5.1.1 and 5.1.2.

Hence, total energy cost per CH for useful work $E_{CH}$ is given by

$E_{CH} = E_{CH-Rx} + E_{CH-Da} + E_{CH-Tx}$ (5.28)

As in [63], each cluster member $j$ consumes $E_{mem}$ energy to transmit to it’s $CH_i$ in a given data transmission round which can be determined as under:

$E_{mem} = l \times E + l \times D_{ji}$ (5.29)

As specified in [63], energy overhead that a CH node $i$ has to spend in a cluster setup phase $E_{CH-oh}$ consists of following energies:

- Energy $E_{CH-can}$ spend in CH candidacy broadcasts

$E_{CH-can} = l \times E + l \times R_i$ (5.30)

where $l$ is the candidacy broadcast packet size.

- Energy $E_{CH-join}$ spent in the reception of join requests from its members

$E_{CH-join} = l \times E \times \left( \frac{n}{k} - 1 \right)$ (5.31)

where $l$ is the packet size of the join request.

- Energy $E_{CH-tdma}$ spent to broadcast the TDMA schedule among its members

$E_{CH-tdma} = l \times E + l \times R_i$ (5.32)

where $l$ is the TDMA schedule message packet length.

Hence

$E_{CH-oh} = E_{CH-can} + E_{CH-join} + E_{CH-tdma}$ (5.33)

Equations 5.29 to 5.33 use notations specified in section 5.1.1 and 5.1.2.
As per [63], overhead energy consumption of the cluster member node $E_{mem-oh}$ during cluster setup phase consists of following energies:

1. Energy $E_{mem\text{-}join}$ required to send the cluster join packet to CH $i$

   $$E_{mem\text{-}join} = l \times E + l \times D_{ji}$$  \hspace{1cm} (5.34)

   where $l$ is the packet size of the join request.

2. Energy dissipation $E_{mem\text{-}can}$ during the CH $i$ candidacy broadcast packet reception.

   $$E_{mem\text{-}can} = l \times E$$  \hspace{1cm} (5.35)

   where $l$ is the candidacy broadcast packet size.

Hence, we obtain $E_{mem\text{-}oh}$ by the following equation

$$E_{mem\text{-}oh} = E_{mem\text{-}join} + E_{mem\text{-}can}$$  \hspace{1cm} (5.36)

### 5.6.1 CLUSTERING

Similar to [60, 112], in the proposed architecture, clustering problem can be formulated as an uncapacitated facility location problem (UFLP) where the objective is to maximize total battery level of all sensor nodes taking into account all kinds of energy consumption. The clustering problem can be formulated as the following integer programming problem:

$$\max \sum_{i \in N} \left\{ b_i - \left( lE + l \sum_{j \in N} D_{ij} y_{ij} + lF_i x_i \right) - lE \sum_{j \in N} y_{ji} - lE_{DA} \sum_{j \in N} y_{ji} \right\}$$  \hspace{1cm} (5.37)

s.t.  \hspace{1cm} $x_i + \sum_{j \in N} y_{ij} + \sum_{k \in N} c_{ik} + \sum_{a \in N} a_{ia} + s_i = 1, \quad i \in N$, \hspace{1cm} (5.38)

$$\left( b_i - \frac{\infty}{n} \sum_{i \in N} b_i \right) x_i \geq 0, \quad i \in N, c_{ik} = 0, a_{ia} = 0$$  \hspace{1cm} (5.39)

$$y_{ij} \leq x_i, \quad i, j \in N, c_{ik} = 0, a_{ia} = 0, c_{ji} = 0, a_{ja} = 0$$  \hspace{1cm} (5.40)

$$x_i \in \{0, 1\}, \quad i \in N,$$  \hspace{1cm} (5.41)

$$y_{ij} \in \{0, 1\}, \quad i \in N, \quad j \in N$$  \hspace{1cm} (5.42)

$$c_{ik} \in \{0, 1\}, \quad i \in N, \quad k \in N$$  \hspace{1cm} (5.43)

$$a_{ia} \in \{0, 1\}, \quad i \in N, \quad a \in N$$  \hspace{1cm} (5.44)

The decision variables used in equations 5.37 to 5.44 are defined above in section 5.1.2.
The aim is to maximize the total amount of remaining batteries of sensor nodes after one round. The constraint (5.38) specifies that each sensor node belonging to ARS $\alpha$ of cell $k$ plays a CH or sends the data to a CH as far as its battery level is positive. Constraint (5.39) ensures that sensor node which has at least $\propto$ times as much as the average battery level of all alive sensor nodes belonging to ARS $\alpha$ of cell $k$ can be the candidate of becoming CH in the next cycle. Constraint (5.40) means that only CHs can receive the data. The objective (5.45) is to maximize the total sum of battery level of sensor nodes. It can be rewritten to the standard form of the objective function in the UFLP similar to [60] as under:

$$\min \sum_{i \in N} \sum_{j \in N} (E + D_{ij} + E_{DA})y_{ij} + \sum_{i \in N} F_i x_i$$ (5.45)

s.t.

$c_{ik} = 0, a_{ia} = 0, c_{jk} = 0, a_{ja} = 0$

5.7 CLUSTER HEAD ROTATION TIMEOUT

The performance of energy driven CH rotation algorithms have shown to be far more superior to time driven CH rotation algorithms when it comes to maximizing the sensor bed lifetime. The sensor bed life time can be improved by selecting the proper points at which a CH role is relinquished to higher energy nodes via a leader election algorithm. As shown in Figure 5.4 and specified in [63], there can be two extreme cases for optimum value for energy threshold '$c$' at which CH rotation should be carried out [63].

- If $c \rightarrow 1$ there will be frequent CH rotations. This allows an even distribution of the CH role among nodes in the WSN but frequent CH rotations would result in considerable energy overhead in control and coordination messages during cluster set up. The overall use of energy for useful work thus will be reduced. Hence, even though the lifetime curve has a sharp edge, the useful lifetime of the WSN is reduced.
- On the other hand when $c \rightarrow 0$ the CH rotations will be less frequent resulting in low overheads. However, now CH nodes would not have enough energy to act as regular nodes after relinquishing the CH role. This would result in a lifetime curve that is less steep.
As observed from [63] an analytical method may be used to identify the optimal point at which a CH rotation has to be initiated.

![Graph showing lifetime of WSN with respect to c](image)

**Fig. 5.4 Lifetime of WSN with respect to c [63]**

Similar to [63], $c_{opt}$ is the value of $c$ which would allow maximizing the sensor bed lifetime. An epoch is defined as the number of data transmission rounds that all nodes in a cluster serve as a CH once. $X_{j,l}$ is the number of data transmission rounds that node$i$ serves as a CH in the $j^{th}$ epoch. $E_{j,l,p}$ is the residual energy level of a $node#p$ at the beginning of the $node#j$’s CH period in the $epoch#j$. Further, it has been assumed that all the nodes have $E_0$ energy at the beginning of the epoch$#0$.

By iteration and referring equations 5.28 and 5.29 as in [63] $X_{j,l}$ can be determined as under:

$$X_{j,l} = \frac{(1-c)E_{j,l}-E_{CH-oH}}{E_{CH}} \quad (5.46)$$

It has been assumed that the $epoch#0$ would deplete its energy during $epoch#j$ when the $node#i$ is acting as the CH. Similar to [63] residual energy of $node#0$ at beginning of $node#i$’s CH role in $epoch#j$, can be given by equation 5.47 using equations 5.29 and 5.33 as below:

$$E_{j,l,0} = cE_{j,0,0} - (i - 1)E_{mem-oH} - (\sum_{k=1}^{i-1} X_{j,k})E_{mem} \quad (5.47)$$
The $p$ represents the total number of data transmission rounds that \textit{node}#0 completes before its total energy depletes. The $p$ consists of two portions $p_1$ and $p_2$. The $p_1$ represents the total number of data transmission rounds for which \textit{node}#0 is alive excluding the data transmission rounds it spend with the previous CH before it dies. The portion $p_2$ is the number of data transmission rounds that \textit{node}#0 spends with it's last CH before it itself dies. Authors in [63] specified methods to calculate the value of $p_1$ and $p_2$ given in equations 5.48 and 5.49 respectively.

\begin{align*}
    p_1 &= \begin{cases} 
        \sum_{p=0}^{j-1} \sum_{q=0}^{N-1} X_{p,q} + \sum_{q=0}^{l-1} X_{j,q} &; j \geq 1 \\
        \sum_{q=0}^{l-1} X_{0,q} &; j = 0
    \end{cases} \quad (5.48) \\
    p_2 &= \begin{cases} 
        \frac{E_{j,0} - E_{memOH}}{E_{memCH}} &; i \neq 0 \\
        \frac{E_{j,0} - E_{CHOH}}{E_{CH}} &; i = 0
    \end{cases} \quad (5.49)
\end{align*}

The total number of data transmission rounds, $p$ can also be represented as a function of variable $c$. Therefore,

\[ p = p_1 + p_2 = f(c) \quad (5.50) \]

The objective is to maximize $p$ with respect to $c$. Hence, when $p \to \max(p)$ and $c \to c_{opt}$, using equation 5.50.

\[ \max(p) = f(c_{opt}) \quad (5.51) \]

\section*{5.8 OPTIMUM NUMBER OF CLUSTERS PER ARS}

Determining optimum number of clusters per ARS is very important. When there is only one cluster, non-cluster head nodes often have to transmit data faraway to reach the cluster head node, draining their energy. Alternatively, when there are many clusters, degree of local data aggregation reduces.

The [24] has given a good method for determining optimum number of clusters per ARS. In a WSN region of $M^2$ consisting of $k$ clusters, area occupied by each cluster would be approximately $\frac{M^2}{k}$. 


101
As in [24], average distance between the member node $i$ and CH $j$ can be given as specified by equation 5.53, assuming WSN area is enclosed in a circle having radius $R$

$$R = \left( \frac{M}{\sqrt{\pi k}} \right) \quad (5.52)$$

$$D_{ij} = \frac{1}{2\pi} \frac{M^2}{k} \quad (5.53)$$

Similarly to [24], energy dissipated in a cluster during the frame $E_{cluster}$, total energy for the frame $E_{total}$ and optimum numbers of clusters $k_{opt}$ can be specified via equations 5.54, 5.55 and 5.56 respectively.

$$E_{cluster} = E_{CH} + \left( \frac{n}{k} - 1 \right) E_{mem} \quad (5.54)$$

$$E_{total} = kE_{cluster} \quad (5.55)$$

$$k_{opt} = \frac{\sqrt{n}}{\sqrt{2\pi}} \frac{\epsilon_{FS} M}{\epsilon_{MP} f_i^2} \quad (5.56)$$

### 5.9 SLEEP STATE TIME

In order to conserve energy, it is desirable to maximize the time, which a sensor node can spend in sleep mode. Each sensor node $i$ sleeps for duration $\text{random}(0, T_i)$ before it wakes up again. As in [113], parameter $T_i$ is specified in equation 5.57:

$$T_i = T \cdot \frac{\text{Energy}_i}{\text{Avg}_\text{energy}_i} \quad (5.57)$$

where $T$ is the duration value in case when all sensors have same energy levels. $\text{Energy}_i$ is the energy level of sensor $i$. $\text{Avg}_\text{energy}_i$ is the average energy level of the neighbours of sensor $i$.

### 5.10 RELIABILITY

The WSNs suffers from high data loss due to the inherent weaknesses in wireless transmission medium, hostile environments, human interferences and node failures. Hence, ensuring data transfer with minimum loss is very important in WSNs. Hongwei Zhang et. al. [114] suggested the window-less block acknowledgement scheme for reliability that improves channel utilization and enables continuous packet forwarding in
the presence of packet and ack-loss. In this approach, the sender S organizes its packet queue as \((M + 2)\) virtual queues, as shown in following Figure 5.5 where \(M\) specifies maximum number of retransmissions at each hop. The virtual queues are ranked such that a virtual queue \(Q_k\) ranks higher than \(Q_j\) if \(k < j\). Virtual queues \(Q_0, Q_1, \ldots \) and \(Q_M\) buffer packets waiting to be sent or to be acknowledged and \(Q_{M+1}\) collects the list of free queue buffers.

![Virtual Queues at Sender](image)

**Fig. 5.5 Virtual Queues at Sender [114]**

As in [114], probability \(P_{ack}\) for receiving the acknowledgement of a packet \(m\) sent by sender \(S\) can be specified via equation 5.58

\[
P_{ack} = 1 - p + \frac{p(1 - 3p + 4p^2 - 2p^3)}{1-p+p^2}
\]

where \(p\) is the probability of losing a single data packet.

### 5.11 CONCLUSION

The parameters pertaining to the various components and aspects of proposed architecture have been studied and analyzed. Their effect on the working of proposed model has also been described. The detailed analysis of energy consumption in various activities of clustering and clustering process has also been explained. The main emphasis in this dissertation has been on designing the new proposed architecture that handles the problem of timely receipt and availability of information to take an effective action in case of crisis. The functionality of the components of the proposed model is based on the standard proven work. Although some alterations have been done according to the needs of the architecture wherever appropriate.
CHAPTER 6
SIMULATION OF MODEL AND RESULT ANALYSIS

6.1 INTRODUCTION

Simulation builds a prototype of a protocol under consideration and helps in the evaluation of its performance through it. It provides the facility to researchers to check functioning of a protocol in different scenarios, configurations and topologies. The simulation has currently become the primary feasible approach for quantitative analysis of the sensor networks. The aim of this chapter is to determine the performance of the proposed architecture in different scenarios in achieving the desired goals.

6.2 SIMULATION OF THE PROPOSED MODEL

It is extremely difficult to analytically model the interactions between all the nodes, for even moderately-sized networks with few nodes. In order to show effectiveness of the proposed model, therefore simulation software has been developed using object oriented language C#. The SQL Server database has been used for storing the results obtained during the running of the simulation software. The sensed events e.g. seismic or environmental are assumed to be localized.

The centralized clustering mechanism has been used in this architecture, in case, the coverage area of a cell is less than or equal to 120m x 120m. In this case single hop communication is used for conveying information from member node to CH as well as from CH to respective destination (ARSs). However, when the sensed area is large, multiple level of clustering has been used within a cell to minimize total energy requirement and for efficiently communicating sensed information over long distances.

The proposed architecture has been compared with three existing protocols LEACH-C for small coverage area, HEED for hierarchical clustering and RAP for deadline miss ratio in large coverage areas by assuming exactly the same environment and assumptions.
6.3 SMALL COVERAGE AREA OF A CELL

The performance of proposed model has been compared with the centralized clustering algorithm LEACH-C by taking similar environment and assumptions as considered by LEACH-C. The network lifetime and scalability are used as a key indicator to evaluate performance of the proposed system. The sensor nodes lose communication ability when their energy is depleted and are considered as “dead”. The data transmissions from sensor nodes were simulated until all the sensor nodes died.

6.3.1 Simulation Setup

Similar to [115], a simple model for the radio hardware energy dissipation has been assumed in which transmitter, power amplifier and receiver dissipates energy to run the radio electronics. Table 6.1 shows simulation parameters for small coverage area of the cell [24].

<table>
<thead>
<tr>
<th>Energy Assumptions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold Transmission distance in meters</td>
<td>d₀</td>
<td>100 meters</td>
</tr>
<tr>
<td>Electronics Energy</td>
<td>E_{elec}</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>Amplifier Energy factor for free space model</td>
<td>E_{fs}</td>
<td>100 pJ/bit/m²</td>
</tr>
<tr>
<td>Amplifier Energy factor for two ray model</td>
<td>E_{tr}</td>
<td>.0013 pJ/bit/m⁴</td>
</tr>
<tr>
<td>Initial Battery Level of all the nodes</td>
<td>InitialBattery</td>
<td>0.5 J</td>
</tr>
<tr>
<td>Aggregation Energy</td>
<td>E_{da}</td>
<td>5 nJ/bit/signal</td>
</tr>
</tbody>
</table>

| Packet Size Assumptions                                                          |   |   |
| Cluster Head proposal Packet Size by Sensor Nodes                               | CHProposal| 800 bits |
| Cluster Head Intimation Packet Size to Sensor Nodes                             | CHIntimation| 400 bits |
| TDMA Schedule Packet Size to Sensor Nodes by Cluster Head                       | TDMA| 800 bits |
| Sensed Message Packet Size by Sensor Nodes                                      | DataPacket| 600 bits |
### Optimum Limits Assumptions Per Round

| Min No. of possible sensed message per node | MaxMessages 2 |
| Max No. of possible sensed message per node | MaxMessages 5 |
| Max No. of possible messages sent by CH | MaxMessagesCH 10 |
| Minimum Level of Battery Required for Node to remain functional | MinBattery 50 nJ/bit |
| Minimum Level of Battery Required for Node to become cluster head | MinBatteryCluster 200 nJ/bit |
| Max No. of possible Clusters Per ARS | MaxClustersPerArs 2 |

#### 6.3.2 Centralized Clustering

In the proposed architecture the centralized clustering algorithm as shown in Table 6.2 has been used.

**Table 6.2 Centralized Clustering Algorithm**

<table>
<thead>
<tr>
<th>Algorithm – Centralized Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
</tr>
<tr>
<td>( N = { \text{Location of all active Nodes} } )</td>
</tr>
<tr>
<td>( A = { \text{Location of higher level Authorities} } )</td>
</tr>
<tr>
<td>( N_a = { \text{Location of all active nodes reporting to Authority } a: a \in A } )</td>
</tr>
<tr>
<td>( C_a = { \text{Location of CH’s connected to Authority } a: a \in A } )</td>
</tr>
<tr>
<td>( N_{ca} = { \text{Location of Nodes belonging to cluster } c:a \in A \land c \in C_a } )</td>
</tr>
<tr>
<td>( C_{\text{count}} = \text{Count of clusters to be formed} )</td>
</tr>
<tr>
<td><strong>Step 0:</strong> Authority ( a \in A ) broadcasts a message for clustering to each node ( n \in N_a )</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Each node ( n \in N_a ) sends its battery Level ( b_n ) to Authority ( a )</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Each Authority ( a \in A ) performs the following procedure</td>
</tr>
<tr>
<td>(2-1) Initialize the Eligibility set of Sensor Nodes</td>
</tr>
<tr>
<td>( E = { \text{Set of nodes } e: e \in N_a \land b_e &gt; MinBatteryCluster \land \forall i,j \in N, i &lt; j \quad b_{e_i} &gt; b_{e_j} } )</td>
</tr>
<tr>
<td>(2-2) Initialize set of Cluster Heads ( C_a ) to 0</td>
</tr>
<tr>
<td>(2-3) if ( (E_{\text{count}} &lt; C_{\text{count}}) )</td>
</tr>
<tr>
<td>( C_a = E )</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>copy first ( C_{\text{count}} ) elements from ( E ) to ( C_a )</td>
</tr>
<tr>
<td><strong>Step 3:</strong> Authority ( a \in A ) broadcasts CHs information ( C_a ) to each node ( n \in N_a )</td>
</tr>
<tr>
<td><strong>Step 4:</strong> For each ( n \in N ) and ( a \in A )</td>
</tr>
<tr>
<td>( N_{ca} = { n: n \in N \land \forall i,j \in C_a, a \in A \land \text{Dist}<em>{ni} &lt; \text{Dist}</em>{nj} } )</td>
</tr>
</tbody>
</table>
The ARS obtains the candidatures of all the nodes connected to it via current CHs. The CHs submits candidature of all members of their own clusters for becoming CH in next round to their respective ARS’s. The ARS accepts the candidatures of only those nodes whose remaining battery level is greater than the minimum threshold battery level. If the eligible nodes are less than the maximum CHs to be chosen than all eligible nodes are chosen as new CHs otherwise top n (max. clusters to be formed) higher candidature level nodes are chosen as new CHs.

In election of new CHs preference has been given to sensor nodes having higher remaining battery level. The proposed algorithm saves the precious energy of sensor nodes, which otherwise they have to spend in deciding among themselves new CHs in many rounds of communication.

6.4 SIMULATION RESULTS FOR SMALL COVERAGE AREA OF THE CELL

The performance of the proposed architecture has been measured on the basis of various parameters like average energy dissipation, system lifetime, successful data delivery, scalability and number of live nodes. To evaluate the proposed architecture, the results were obtained by varying number of sensor network characteristics like number of cells and sensor nodes within a cell.

6.4.1 Single Cell containing 48 Sensors

First of all, a single cell having 48 sensor nodes in a cellular region of 120 m x 120 m has been considered. The nodes are distributed to cover every region of the area under consideration. The results obtained as a result of simulation process are shown in Figures from 6.1 to 6.4. The simulation process has been simplified by taking maximum of 12 clusters per cell and 2 clusters per ARS, although performance may be improved further by taking optimum number of clusters [63].
Fig. 6.1 Remaining Average Battery Level per Node over various Rounds

Fig. 6.2 Number of Live Nodes over various Rounds
Fig. 6.3 Number of Messages Received over various Rounds

Fig. 6.4 Number of Messages received per ARS over various Rounds
**Discussion and Analysis**

The LEACH-C although appears to be a promising centralized clustering algorithm, there are few areas left for making protocol more energy efficient. The proposed architecture is more energy efficient compared with LEACH-C as shown in Figures 6.1 and 6.2. It is because average distance between cluster member nodes and respective ARSs, is much shorter than average distance between sensor nodes of clusters and the single base station, in case of LEACH-C.

The longevity of the model is also more than LEACH-C as shown in Figure 6.3 and 6.4. In LEACH-C all the nodes are dead in after nearly 800 rounds where as in case of this model the nodes continue to work until around 1100 rounds. The number of messages received by the model as shown in Figures 6.5 and 6.6 are also much higher as compared to LEACH-C.

**6.4.2 Single Cell containing 72 Sensors**

The performance of the proposed model is measured, considering a single cell having 72 sensor nodes in a cellular region of 120 m x 120 m. The results of the simulation process are shown in Figures from 6.5 to 6.8. The results were obtained by assuming maximum of 18 clusters per cell and maximum of 3 clusters per ARS.
Fig. 6.5 Number of Messages Received over various Rounds

Fig. 6.6 Number of Messages received per ARS over various Rounds
Fig. 6.7 Number of Live Nodes over various Rounds

Fig. 6.8 Average Battery Consumed per Node over various Rounds
Discussion and Analysis

The total number of messages received over various rounds per cellular area is directly proportional to the number of nodes. In this model and LEACH-C, as shown in Figure 6.5 and 6.6, more number of messages are received over various rounds due to increase in the number of nodes.

The nodes which are nearer to the ARSs die later since amount of energy dissipation is directly proportional to the distance. The network lifetime is therefore slightly improved in the proposed model as compared to LEACH-C due to addition of more number of nodes near the various ARSs as shown in Figure 6.7. The average battery consumption per node as shown in Figure 6.8 also remains same in proposed model and LEACH-C.

6.4.3 Single Cell containing 96 Sensors

The performance of the proposed model is measured by considering a single cell having 96 sensor nodes in a cellular region of 120 m x 120 m. The results are shown through Figures 6.9 to 6.12. The results were obtained by assuming maximum of 24 clusters per cell and maximum of 4 clusters per ARS.
Fig. 6.9 Average Battery Consumed per Node over various Rounds

Fig. 6.10 Number of Messages Received over various Rounds
Fig. 6.11 Number of Live Nodes over various Rounds

Fig. 6.12 Number of Messages received per ARS over various Rounds
Discussion and Analysis

The average energy dissipation over various rounds in both the proposed architecture and LEAH-C, remains constant in case of single cell and does not depends upon the number of nodes as shown in Figure 6.1, 6.8 and 6.9. This is because

(i) Both models use centralized clustering where various CHs relay aggregated messages to their respective destinations using 1-hop communication.
(ii) The member nodes also use 1-hop communication to submit their readings to their current cluster head.
(iii) All nodes submit their candidature for becoming CH to their respective parent after every round.
(iv) The average distance of nodes from respective destinations remains constant.

The number of messages received over various rounds and ultimate network lifetime although is dependent on number of nodes as shown in Figures 6.2-6.7 and 6.10-6.12.

6.4.4 Two Adjacent Cells containing 192 Sensors and 11 ARS

The results were obtained to check the scalability of the proposed architecture by considering two adjacent cells of 120 m x 120 m and each having 96 sensor nodes. In both the cells collectively, a total of 11 ARSs were taken with one common ARS. The maximum of 24 clusters per cell and 4 clusters per ARS were assumed for analysis.

The information received by ARSs lying on the edge of two cells is replicated in the Base-Stations of each cell to improve fault tolerance. So, that in case of collapse of one base station the requisite information can be obtained from the adjacent cell’s base station. The results are shown through Figures 6.13 to 6.17.
Fig. 6.13 Average Battery Consumed per Node over various Rounds

Fig. 6.14 Number of Messages received per ARS over various Rounds
Fig. 6.15 Number of Messages per Cell over various Rounds

Fig. 6.16 Number of Messages Received over various Rounds
Fig. 6.17 Number of Live Nodes over various Rounds

Discussion and Analysis

The results in Figure 6.13, shows that the proposed model is more scalable than LEACH-C since the network lifetime almost remain constant, even if coverage area is increased. The main reason for this is the use of distributed approach by the proposed architecture where load of clustering and other activities is evenly distributed on various ARSs.

The number of sensed messages as shown in results from Figure 6.14 to 6.16, as well as number of alive nodes as shown in results from Figure 6.17, increases expectedly in the proposed architecture as a result of more sensor nodes, since its performance is not degraded as numbers of cells are increased.
6.4.5 Three Cells containing 288 Sensor Nodes and 15 ARS

The more exhaustive results were obtained, taking into consideration three adjacent cells of 120 m x 120 m and each having 96 sensors to check the scalability of the proposed architecture. All the cells collectively consist of 15 ARSs with three common ARSs and 288 sensor nodes. The results are shown in Figures 6.18 to 6.22.

![Graph showing number of messages received per ARS over various rounds](image)

**Fig. 6.18 Number of Messages received per ARS over various Rounds**
Fig. 6.19 Number of Messages per Cell over various Rounds

Fig. 6.20 Number of Messages Received over various Rounds
Fig. 6.21 Remaining Average Battery Level per Node over various Rounds

Fig. 6.22 Number of Live Nodes over various Rounds
Discussion and Analysis

The performance of LEACH-C degrades with the increase in number of cells whereas performance of the proposed architecture almost remains unaffected, as observed in results from Figure 6.18 to 6.22. The proposed architecture although uses additional ARS components but besides relaying information to BS, they are also required to perform comprehensive and effective action in the covered area, in case of crisis situations.

6.4.6 Four Cells containing 392 Sensor Nodes and 19 ARS

The more results have been obtained taking into account four adjacent cells of 120 m x 120 m and each having 96 sensors to verify the performance of the proposed architecture. The cells collectively consist of 19 ARSs with five common ARSs and 392 sensor nodes. The results are illustrated in Figures from 6.23 to 6.27.

Fig. 6.23 Number of Messages per Cell over various Rounds
Fig. 6.24 Number of Messages received per ARS over various Rounds

Fig. 6.25 Number of Messages Received over various Rounds
Fig. 6.26 Remaining Average Battery Level per Node over various Rounds

Fig. 6.27 Number of Live Nodes over various Rounds
Discussion and Analysis

The role of ARSs is extremely important in the proposed model since besides taking automatic action in case of disaster. They play many other roles like replicating the sensed information, coordinating with nodes, clustering after every round and collaboration with other ARSs. The proposed architecture as shown by results from Figure 6.1 to 6.27 is better as compared to LEACH-C for designing WSAN based surveillance applications that requires small area to be sensed like within a building only.

6.5 SIMULATION FOR LARGE COVERAGE AREA OF THE CELL

In case area to be monitored by WSN is large, single hop communication is not suitable due to higher energy requirements. The hierarchical clustering mechanism and multi hop communication has been used in this case, for handling the peculiarities of large area sensor network.

The higher level CHs forwards data gathered from the sensor nodes in their respective clusters as well as data gathered from lower level CHs towards the destination. Hence, the CHs form a network among them and use multi-hop paths for routing data to their respective ARSs. The one-hop routing can be considered as a special case of multi-hop routing, where each CH node transmits its data directly to the respective ARS, i.e., the ARS is the next-hop neighbour for every CH node in the network.

In the proposed architecture, various clusters are formed in a top-down manner. In this protocol, once clusters have been organized, the CHs form a multi-hop routing backbone. In multi-hop routing, the data packets received from the source node are relayed through the intermediate nodes until they reach the destination in order to reduce the transmission energy consumption.
For intra-cluster communication, every member node forwards the data to the CH directly whereas for sending the data towards destination, inter-cluster hierarchical routing is adopted to decrease the latency. The data travels from a lower clustered layer to a higher one. Although, it hops from one node to another, but as it hops from one layer to another it covers large distances.

This approach is much better than traditional multi-hop routing between nodes. Further, in cluster based models, only cluster heads perform data aggregation whereas in the traditional multi-hop model, every intermediate node performs data aggregation. As a result, cluster based model is more suitable for time critical applications than the traditional multi-hop models. The proposed architecture was evaluated on two different aspects – network lifetime estimation and deadline miss ratio.

The performance of proposed model has been compared for network lifetime estimation with popular Hierarchical-Clustering algorithm HEED. The LEACH approach will be inappropriate in this scenario because this model assumes that all CHs only communicate directly with the Base-Station. The proposed architecture has been compared with RAP, for checking its effectiveness in achieving better deadline miss ratio.

6.5.1 Simulation Setup

A model for radio hardware energy dissipation [24] has been assumed, in this model, the transmitter, the power amplifier and the receiver dissipates energy to run the radio electronics. The simulation was performed by considering similar parameters as taken in [24]-[26] in case of large scale sensor networks. Table 6.3 highlights simulation parameter taken to run simulation process for large coverage area of the cell.
Table 6.3 Simulation Parameters for large coverage area of the cell

<table>
<thead>
<tr>
<th>Energy Assumptions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics Energy E_{elec}</td>
<td>50 nJ/bit</td>
<td></td>
</tr>
<tr>
<td>Amplifier Energy E_{fs}</td>
<td>100 pJ/bit/m²</td>
<td></td>
</tr>
<tr>
<td>Initial Battery Level of all the nodes InitialBattery</td>
<td>5 J</td>
<td></td>
</tr>
<tr>
<td>Aggregation Energy E_{da}</td>
<td>5 nJ/bit/signal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Packet Size Assumptions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensed Message Size SendSensMessSize</td>
<td>260 bits</td>
<td></td>
</tr>
<tr>
<td>Candidature submission message size CandSubMessSize</td>
<td>280 bits</td>
<td></td>
</tr>
<tr>
<td>Cluster Head Intimation CHIntimation</td>
<td>100 bits</td>
<td></td>
</tr>
<tr>
<td>TDMA schedule TDMA</td>
<td>100 bits</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimum Limits Assumptions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min No. of possible sensed message in a round per node MaxMessages</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Max No. of possible sensed message in a round per node MaxMessages</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Nodes per Cluster NodesPerCluster</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Minimum Level of Battery Required for Node to remain functional MinBattery</td>
<td>50 nJ/bit</td>
<td></td>
</tr>
<tr>
<td>Minimum Level of Battery Required for Node to become cluster head MinBatteryCluster</td>
<td>200 nJ/bit</td>
<td></td>
</tr>
<tr>
<td>Sensing Radius Node SensRadius</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>Aggregation Timer (Milli Seconds) AggrTimer</td>
<td>100 ms</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>20 s</td>
<td></td>
</tr>
<tr>
<td>Delay Time (Milli Seconds)</td>
<td>5 ms</td>
<td></td>
</tr>
<tr>
<td>Message Velocity</td>
<td>500 m/s</td>
<td></td>
</tr>
</tbody>
</table>

6.5.2 Hierarchical Clustering

The hierarchical clustering algorithm has been used as shown in Table 6.4. This algorithm produces clusters that improve the lifetime of the network.
### Table 6.4 Hierarchical Clustering

**Algorithm - Hierarchical Clustering**

**Input:**
- \( N = \{\text{Location of all active Nodes}\} \)
- \( A = \{\text{Location of all ARSs}\} \)
- \( C_a = \{\text{Location of all CH's connected to ARS a: } a \in A\} \)
- \( L_c = \{\text{Level No. of all CH's: } c \in C_a\} \)
- \( P_c = \{\text{Location of parent CH of CH c: } c \in C_a \land \text{Level}_{P_c} < \text{Level}_c \land \text{Dist}_{P_c} \leq \text{SensingRadius}\} \)
- \( S_c = \{\text{Location of subordinates CHs of CH c: } c \in C_a \land \text{Levels}_c > \text{Level}_c \land P_{S_c} = c\} \)
- \( M_c = \{\text{Set of member Nodes m belonging to cluster: } m \in N \land c \in C_a\} \)
- \( N_{sr} = \text{Sensing Radius of Node,} \)
- \( M_{bc} = \text{Minimum required battery for becoming CH, } b_n = \text{Battery of Node n} \)

**Step 1:** Repeat steps 2–4 for each ARS \( a \in A \)

**Step 2:** set \( C_{\text{level}} = 0 \)

**Step 3:** Each CH \( c \in C_a \) having \( L_c = C_{\text{level}} \) performs the following procedure:

1. **Broadcasts a message for submission of candidature to each member node \( n \in M_c \)**
2. **Each member node \( n \in M_c \) sends its weight \( w_n \) to CH \( c \)**
3. **CH \( c \in C_a \) determines the New Leader \( l \)**
   - \( l = \{e: e \in M_c \land b_e > M_{bc} \land \forall i, j \in M_c, i < j \Rightarrow w_{e_i} \geq w_{e_j} \land \text{Dist}_{e_iP_c} \leq N_{sr} \land \text{Dist}_{e_jS_c} \leq N_{sr}\} \)
4. **The member node \( l \in M_c \) becomes new parent of current cluster and sub‐ordinate CHs \( c = l \)**
   - \( P_{S_c} = l \)
5. **set \( C_{\text{level}} = C_{\text{level}} + 1 \)**

**Step 4:** Each non‐CH nodes \( n \in N \) joins nearest CH \( i \in C_a \)

- \( M_i = \{n: n \in N \land \forall i, j \in C_a, a \in A \Rightarrow \text{Dist}_{ni} < \text{Dist}_{nj}\} \)

**Step 5:** Determine next orphan node \( o \in \{o: o \in N \land o \notin M_c \land o \notin C_a\} \)

**Step 6:** If no more orphan node found then return

**Step 7:** If orphan Node \( o \) is eligible for becoming Level 0 CH

1. **e = \{o: o \in N \land o \notin M_c \land o \notin C_a \land \text{Dist}_{oa} \leq N_{sr} \land a \in A \land b_o > M_{bc}\}**
2. **set \( L_e = 0 \)**
3. \( C_a = C_a \cup e \)
4. **Unconnected nodes \( u \in N \) joins new CH \( e \)**
   - \( N_u = \{u: u \in N \land u \notin M_c \land u \notin C_a \land \text{Dist}_{ue} \leq N_{sr}\} \)
5. **Go to Step 5**

**Step 8:** If target node \( t \) for orphan node \( o \) exists

1. **t = \{i: i \in M_c \land \text{Dist}_{oi} \leq N_{sr} \land \text{Dist}_{ia} \leq \text{Dist}_{oa} \land a \in A \land \forall i, j \in N, \text{Dist}_{ja} \geq \text{Dist}_{ia} \land b_i > M_{bc}\}**
2. **set \( L_t = \{L_c: t \in M_c \land c \in C_a\} \)**
3. \( C_a = C_a \cup t \)
4. **Unconnected nodes \( u \in N \) joins new CH \( t \)**
   - \( N_u = \{u: u \in N \land u \notin M_c \land u \notin C_a \land \text{Dist}_{ut} \leq N_{sr}\} \)

**Step 9:** Go to step 5
**Working of algorithm**

In the proposed architecture the clustering is done in a top down manner. First of all top level CHs are chosen by the ARS. The current top level CHs collects candidature weight-age from their member nodes. Election is done on the basis of the weight-age of the nodes candidature.

The current CHs choose the member node as a new leader that is connected to all the lower level CHs and has highest weight. This information is then broadcast to lower clusters. The next level CHs then repeats this process till last level is reached. All the non-CH nodes then join the nearest CH.

If some sensor nodes fails to connect to any CH, then first of all the ARS chooses few CH’s among them and connects rest with them. The remaining unconnected nodes elect a non-CH node at higher level as their new parent and becomes it member. The process continues until maximum possible unconnected nodes are connected.

### 6.6 SIMULATION RESULTS FOR NETWORK LIFETIME ESTIMATION FOR LARGE COVERAGE AREA OF A CELL

The performance of the proposed architecture has been measured, considering a single cell having 1634 sensor nodes, for representation of every portion of a cellular region of 1200m x 1200m. The proposed architecture was compared with the hierarchical clustering algorithm HEED taking exactly the same environment and assumptions. The purpose being to show its effectiveness in achieving energy efficiency, on various parameters like averages energy consumption per node, total messages received by Base Station and average energy consumption of nodes having Base Station within their sensing Radius. The results can be seen in Figures from 6.28 to 6.30.
Fig. 6.28 Average Battery Consumed per Node over various Rounds

Fig. 6.29 Number of Messages Received over various Rounds
Fig. 6.30 Average Battery Consumption of Nodes having BS within Sensing Radius

Discussion and Analysis

As shown in Figure 6.28, the proposed architecture consumes lesser energy per node over various rounds as compared to HEED architecture because it uses load balanced and fully connected multi-level clustering architecture. In addition to this, compared to HEED, in the proposed architecture, all lower level clusters are within the sensing range of parent CH, therefore there is no need to use higher power levels to send the sensed information to parent CH. The CHs are changed after every round of operation to further increase the longevity of the network. Further, in this proposed architecture, the residual energy and communication cost factors have been used for choosing CHs instead of using complex calculations.
As illustrated in Figure 6.29, this architecture receives more messages per round compared with HEED. In the proposed architecture, current CHs chooses leader for next round, instead of letting low energy sensor nodes deciding the new leader among themselves in many rounds of communication and spending precious energy. It results in better energy utilization and faster clustering.

In the case of HEED, if a node becomes CH such that BS is outside its sensing range and no CH is nearer to BS than it, all the messages received by it will be lost since there is no way of sending sensed or received messages to BS. This situation is avoided in the proposed architecture because a node becomes a CH only if it is connected to parent CH and ultimately to its respective destination.

The sensor network will stop working if all the nodes which have BS within their sensing radius are dead. In this case, there will be no path to send sensed messages to BS. In the proposed architecture, there are multiple levels 0 CHs connected to ARS therefore; the load of relaying messages is equally distributed. The ARS judiciously chooses level 0 clusters so that clusters are well distributed and load balanced instead of arbitrarily forming of clusters as in case of HEED. As a result, as shown in Figure 6.30, this algorithm performs better than HEED in enhancing the longevity of the network.

6.7 SIMULATION RESULTS FOR DEADLINE MISS RATIO FOR LARGE COVERAGE AREA OF A CELL

The performance of the proposed architecture, in achieving energy efficient real time communication has been compared with real time protocol RAP. In the proposed architecture, like RAP, the Velocity Monotonic Scheduling (VMS) has been used for determining the local urgency at each level/hop, when packets with same deadline have different distances to their destinations. The RAP algorithm performs local aggregation of sensed messages and relays the aggregated information to BS, using multi-hopping from one node to another. In RAP, the nodes relay received messages according to their priority determined by VMS.
Taking same environment and assumptions, performance of both the algorithms have been compared, in terms of various factors like messages received by BS, percentage of messages achieving their deadline, and lifetime of the network over various rounds of communication. The performance was judged by varying number of cluster formation characteristics like changing cluster head after every round, changing CH only after its battery level falls below a threshold value or changing CH election weight-age.

6.7.1 Changing CH after every round and choosing remaining battery level as major factor for electing CH

The performance has been measured considering a single cell having 1634 sensor nodes in a cellular region of 1200 m x 1200 m. In the proposed architecture in this methodology, all the nodes submit their candidature to the current CH after every round. The CH then elects a member node as a new leader for next round, if it has the connectivity with parent and child CHs as well as highest remaining battery level. The results are shown in Figures 6.31 to 6.35.

![Graph showing percentage of messages achieved deadline](image)

Fig. 6.31 Percentage of Messages which have achieved Deadline
Fig. 6.32 Average Number of Hops per Message

Fig. 6.33 Average Battery Consumed per Node over various Rounds
Fig. 6.34 Remaining Battery Level of Nodes having BS within Sensing Radius

Fig. 6.35 Number of Messages over various Rounds
Discussion and Analysis

The proposed architecture achieves better deadline ratio than RAP, as shown in Figure 6.31, because it uses a cluster based hierarchical routing to send the messages towards the destination for decreasing the latency. The data travels from a lower clustered layer to a higher one. Although, it hops from one node to another, but when it hops from one layer to another it covers large distances. In the proposed architecture number of cluster levels required to cover all the nodes, as shown in Figure 6.32, are much lesser than the number of hops each message has to travel as in case of RAP, therefore sensed information reaches the destination quickly. It also results in lower energy consumption as compared to RAP as shown in Figure 6.33.

In the proposed architecture, each CH/node also knows its destination thus they can quickly send the messages towards them as no precious time is wasted in determining the best destination. In RAP architecture, precious information may be lost if it reaches a node which has BS outside its sensing radius and has no node nearer to the BS than it. The sensor network becomes dysfunctional if all the nodes which have connectivity to the BS are dead. The RAP although achieves the goal of real time communication to a great extent but in this model there is huge pressure on nodes nearer to BS, as a result as shown in Figure 6.34, their energy dissipates very quickly and after that there is no way that any message can reach BS. The proposed architecture therefore as illustrated in Figure 6.35 receives more messages over various rounds as compared to RAP.

6.7.2 Changing CH after every round and choosing distance from controlling authority as major eligibility factor for electing CH

The performance has been measured taking into account a single cell having 1634 sensor nodes in a cellular region of 1200 m x 1200 m. In the proposed architecture, all the nodes submit their candidature to the current CH after every round. The CHs in this methodology, elects a member node as a new leader for next round, if it has connectivity with parent and child CHs as well as having highest distance from destination. The results are shown in Figures from 6.36 to 6.40.
Fig. 6.36 Percentage of Messages which have achieved Deadline

Fig. 6.37 Average Number of Hops per Message
Fig. 6.38 Number of Messages over various Rounds

Fig. 6.39 Average Battery Consumed per Node over various Rounds
Discussion and Analysis

More number of messages achieves deadline miss ratio, as shown in Figure 6.36, on choosing distance from ARS as major factor for determining eligibility of member node as a new leader in next round. It is due to the fact in this case respective messages reaches destination faster as they have to travel lesser distance.

The average number of hops each message has to travel in order to reach the destination is also reduced, as revealed in Figure 6.37. The number of messages received over various rounds, as illustrated in Figure 6.38, also increases, since total CHs required to cover the entire area becomes lesser. As a result, more number of nodes performs the task of worker (i.e. sensing the environment for data) instead of acting as a manager (working as a CH).
The main drawback of choosing distance from ARS, as a major factor for electing member node as CH in next round, is higher energy dissipation of sensor nodes, as shown in Figure 6.39, thereby shortening the lifetime of the network. It also exhausts the battery level of sensor nodes having direct connectivity with the destination (ARS) even more quickly, as illustrated in Figure 6.40. As a result, the WSN becomes dysfunctional speedly, since when all such nodes are dead there is no way any message can be sent towards destination.

The weight-age factor for choosing Cluster Heads for next round may be varied dynamically by the ARS depending upon either of the following two scenarios:

- In case of disaster or crisis, the weight-age can be on the basis of distance from destination so that more and quicker information from the event area can be obtained.
- In case of lean period, the weight-age can be on the basis of remaining battery level of sensor nodes so that lifetime of the network can be increased.

6.7.3 Changing CH only after its battery level falls below a threshold value and choosing distance from destination as major factor for electing CH

The performance has been measured in view of a single cell having 1634 sensor nodes in a cellular region of 1200 m x 1200 m. In this methodology, instead of changing the CHs after every round of operation, the current CH once chosen continues to work until its battery level falls below a specified threshold level.

Once its battery level falls below minimum battery level, the current CH relinquishes the management of the cluster and starts working like an ordinary node (may be for the remaining lifetime of the network). In such case, existing member node of the cluster which has connectivity with the parent and subordinate CHs as well as highest weight-age is elected as the new leader. The results are illustrated in Figures 6.41 to 6.45.
Fig. 6.41 Average Battery Consumed per Node over various Rounds

Fig. 6.42 Percentage of Messages which have achieved Deadline
Fig. 6.43 Average Number of Hops per Message

Fig. 6.44 Number of Messages over various Rounds
Discussion and Analysis

This approach is much better than the previous chosen approaches for the proposed architecture. It is because in order to elect the new leader after every round, the precious energy of nodes is saved, as they are not required to spend energy in sending and receiving many messages. The average remaining battery of nodes is therefore enhanced thereby increasing the lifetime of the network as revealed in results of Figure 6.41.

The distance from the destination will be the major factor in determining CHs since initially all the nodes are assumed to have same battery level therefore it will result in much better percentage of messages meeting their deadline miss ratio as shown in Figure 6.42. The percentage of messages meeting their deadline miss ratio will also remain constant over number of rounds since the current elected leaders continues in their capacity. Once the battery of current leader is exhausted the member node of current cluster which has connectivity with both parent and subordinate CHs is chosen as the new leader.
The distance of the new leader from destination, number of cluster levels and average number of hops a message travels to reach destination almost remains constant, as shown in Figure 6.43. In this methodology, minimum clusters will be formed as compared to previous approaches and therefore more nodes will perform sensing duty thereby more messages as shown in Figure 6.44 will reach the destination.

The only drawback of this approach is that the battery level of sensor nodes having destination within their sensing radius is rapidly exhausted, as illustrated in Figure 6.45. Assuming that all the sensor nodes have uniform power level, the sensor network becomes ineffective a little bit rapidly although overall average remaining battery level of sensor nodes will still be higher.

This situation although can be somewhat diluted if we assume that sensor nodes can communicate using multiple power levels. They can use higher power level to send messages to longer distances in order to reach destination. After some time, the sink may notice the void situation and intimates the controlling authority to drop more nodes in the dead zone.

6.8 SIMULATION RESULTS FOR FAULT TOLERANCE

The connectivity refers to the ability of active nodes to stay connected. Designing and deploying a sensor network with considerations about connectivity will provide fault tolerance without the need for fault detection or recovery functionality. The proposed architecture realizes that a fault-tolerant operation is critical to the success of WSN.

In a cell there are six ARS although only one ARS within cell is enough to convey information of all CHs to the BS. The proposed architecture therefore will not become dysfunctional even if one or more ARSs within a cell stops working due to hostile conditions. In the proposed architecture, it is assumed that all the nodes maintain information about the known ARSs and their status.
The information is obtained, when the sensor node receives invitation messages for joining cluster that were broadcasted by various CHs from time to time. The received invitation message contains the location of the ultimate destination (i.e. ARS). The sensor node since knows its position and position of alternative destinations, in case of failure of the current destination due to any eventuality, it sends the join request to the CH that is connected to the second nearest ARS. The results are shown in Figures from 6.46 to 6.49.

**Fig. 6.46 Messages received per Base-Station**
Fig. 6.47 Average Remaining Battery of nodes

Fig. 6.48 Average Battery Consumption per Node
**Fig. 6.49 Avg. Battery consumption of nodes within sensing radius of destination**

**Discussion and Analysis**

The proposed architecture is designed to operate under adverse conditions. As the results shows the number of messages received by the base station almost remains constant even if one or more ARSs are destroyed due to any unavoidable crisis, as shown in Figure 6.46.

Some of the messages although have to travel greater distance in order to reach new destination in case some ARSs are destroyed. Here, higher distance means higher dissipation of energy because energy overhead is directly proportional to the communication distance.

The load on remaining CHs which have destination within their sensing radius (Level 0 CHs) also increases as they have to handle extra traffic thereby increasing their energy consumption and reducing the lifetime of the network, as shown in Figure from 6.47 to 6.49. In the crisis situations, although continued working of the network is more important than longer lifetime.
6.9 CONCLUSION

The proposed architecture has been compared with three existing protocols LEACH-C for small coverage area, HEED for hierarchical clustering and RAP for deadline miss ratio in large coverage areas. The performance was measured by altering number of nodes per cell and total cells in case of small coverage area. The proposed architecture is more energy efficient, increases system lifetime, scalable and receives more messages over various rounds as compared to LEACH-C.

In case of large coverage area, the hierarchical clustering technique has been used. The proposed architecture consumes lesser energy per node and transmits more messages over various rounds as compared to HEED architecture. As a result, it enhances the longevity of the network.

Further, the Velocity Monotonic Scheduling (VMS) has been used for real time communication in the proposed architecture for determining the local urgency at each level/hop.

The performance was judged by varying number of cluster formation characteristics like changing cluster head after every round, changing CH only after its battery level falls below a threshold value or changing CH election weight-age. The proposed architecture achieves better deadline ratio than RAP besides being more energy efficient than it.

The proposed architecture is designed to operate under adverse conditions. As the results show the number of messages received by the base station almost remains constant even if one or more ARSs are destroyed due to an unavoidable crisis.
The proposed architecture in essence provides the following benefits:

i. It works efficiently even in case of hostile situations.

ii. It is scalable.

iii. No messages reach dead-end since it is based on connected approach.

iv. It is more energy efficient since clustering is done in a distributed fashion by ARSs and CHs.

v. It provides higher percentage of deadline miss ratio by not compromising on the lifetime of the network.

vi. It is more fault-tolerant as destruction of some BSs or ARSs will not render it dysfunctional.
CHAPTER 7
CONCLUSION AND FUTURE DIRECTIONS

The WSN technology permits monitoring of the ambient environment at an economical cost much lower than currently possible. The sensor network provides a robust service in hostile environments where human participation is too dangerous. They detect and report the fine grain temporal and spatial dynamics of monitored variables across a landscape. The energy is a critical factor in order to extend the lifetime of the network, since nodes once deployed cannot be recharged. In order to reduce energy consumption, the clustering is a standard approach. It results in significant reduction of energy consumption of sensor nodes. It also helps in achieving efficient and scalable control.

In case of crisis situations, automatic action in the target area is required. The action should also be performed in a timely manner otherwise it will be useless. Effective decision making especially in a crisis situation largely depends upon immediate access and interpretation of local information within the context of the overall environment at any particular point of time. Ensuring data availability is therefore very important especially in situations where there is a high risk of collapse of various components of WSN due to any inevitable disaster or enemy attack. In case the only storing and controlling data entity is destroyed it will result in uselessness of the entire network. The data must be stored in a decentralized manner otherwise it will result in higher energy dissipation as well as inferior and sluggish data retrieval which can affect the mission critical applications.

7.1 CONCLUSION

The problem of timely receipt and availability of information to take an effective action in case of disasters or enemy attack was explored in this dissertation. In disaster management in case of crisis rapid response is required. Moreover, the collected and delivered sensor data must be valid at the time of action. The issue of timely and
enhanced availability besides energy saving therefore is extremely important for effective and automatic response in a real world mission critical WSN based applications.

A new architectural framework has been proposed in this dissertation based on distributed cellular architectural framework to provide various features like timeliness guarantee, fault-tolerance, data integrity, data-centric and distributed storage, besides enhanced energy efficiency. In the proposed model, sensor nodes are deployed to sense information and sensed information is stored. In case of emergency stored information is queried for dynamic decision making and taking automatic actions on the basis of it. The proposed architecture is expected to act as a base for designing practical applications especially in the area of crisis management.

In this dissertation working of various components and aspects of the proposed architecture are specified in detail. The architecture is designed to work in different environments and crisis situations. Depending upon the area of ambient environment to be monitored, different clustering mechanisms are used for improving the estimated network lifetime. The centralized clustering approach and one hop communication has been used for energy efficiency, scalability and timeliness whenever area to be monitored is small. The hierarchical clustering mechanism has been used, for handling the peculiarities of large area sensor network and achieving requisite deadline miss ratio. The architecture has been designed to work in adverse situations and can bear the loss of some critical components of the WSN.

The proposed architecture enhances the availability of data by using replication mechanism. It stored data can also be analyzed to prevent the attacks or disasters in future. The proposed architecture avoids the problem of low energy sensor nodes becoming CHs like in other clustering approaches, resulting in message loss if in between a round, their energy is dissipated. It also achieves better deadline miss ratio without compromising on lifetime of the network. Moreover, performance of the proposed architecture do not degrades if monitored area is enhanced. It also reduces loss of
messages if they reach a node which has BS outside its sensing radius and has no node nearer to BS than it as in case of many other approaches.

Simulation results have been analyzed to evaluate the performance of the proposed model on the basis of factors like average energy dissipation, system lifetime, number of messages received, successful data delivery, scalability, deadline miss ratio and fault tolerance. The performance was judged by varying number of sensor network characteristics like number of cells, number of nodes within a cell etc.

In evaluation of the proposed model, centralized clustering algorithm has been used in case of small coverage area and results are compared with centralized clustering algorithm LEACH-C. In case of large coverage area, multi-level clustering has been used and obtained results have been compared with hierarchical algorithm HEED for network lifetime estimation.

It has also been compared with real time protocol RAP for determining its effectiveness in achieving energy efficient real time communication. Results confirmed that the proposed architecture achieves better deadline miss ratio and enhances lifetime of the WSN. The proposed architecture is designed to operate under adverse conditions. It will continue to work unabatedly even if some critical components of the model are destroyed in the crisis situation.

7.2 FUTURE DIRECTIONS

The cost of human life is very important that cannot be compensated by anything. The proposed architecture will be extremely useful in designing practical applications especially in crisis management. Larger tests on the actual model of the proposed architecture are still required before its actual deployment in the real world. There are still several research challenges that need to be addressed before its use in real deployment. The following points of reference therefore for further work are identified.
• Better localization techniques are needed since existing techniques do not provide enough accuracy in WSNs.

• Looking forward, one can expect that a lot more power-efficient designs will be produced from this framework.

• Although limited mobility can be supported by using GPS System; it is still an open issue in the WSN.

• Larger tests or a mathematical model is needed to determine the optimal transition period to the change the number of clusters in the network. Message loss in the sensor network is another problem that needs to be looked at.

• Choosing appropriate database and DBMS to be used for base station
## ANNEXURE A

### DATA STRUCTURES USED IN PROPOSED MODEL

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
</table>
| 1.   | typedef Location {
| 2.   |   int xLocation;
| 3.   |   int yLocation;
| 4.   |   };
| 5.   |   |
| 6.   | typedef Time {
| 7.   |   int Hours;
| 8.   |   int Minutes;
| 9.   |   int Seconds;
| 10.  |   };
| 11.  |   |
| 12.  | typedef CoverArea {
| 13.  |   Location StartLocation;
| 14.  |   Location EndLocation;
| 15.  |   };
| 16.  |   |
| 17.  | typedef Reading {
| 18.  |   int Value;
| 19.  |   CoverArea Area;
| 20.  |   Time ReadingTime;
| 21.  |   Location SenderID;
| 22.  |   };
| 23.  |   |
| 24.  | typedef ResultSet {
| 25.  |   Location SenderID;
| 26.  |   Reading Reading[];
| 27.  |   };
| 28.  |   |
| 29.  | typedef ParameterInfo {
| 30.  |   int ComponentType;
| 31.  |   int Code;
| 32.  |   int Value;
| 33.  |   };
| 34.  |   |
| 35.  | typedef ElectionInfo {
| 36.  |   Location SenderID;
| 37.  |   int EnergyLeft;
| 38.  |   int DistanceFromDestination;
| 39.  |   };
| 40.  |   |
typedef Message {
    Location SenderID;
    Location DestinationID;
    int MessageType;
    int MessageData;
};

typedef UpdateParameterMessage {
    Location SenderID;
    Location DestinationID;
    ParameterInfo Parameter;
};

typedef Schedule {
    Location SenderID;
    int Schedule;
};

typedef Query {
    Location SenderID;
    Location DestinationID;
    CoverArea Area;
};

typedef ArrayNode {
    Location locNode;
};

typedef ArrayCluster {
    Location locCH;
    ArrayNode locNodes[3];
};

typedef ArrayActionUnit {
    Location locARS;
    ArrayCluster locClusters[2];
};

typedef ArrayCell {
    Location locBS;
    ArrayActionUnit locActionUnits[6];
};

typedef SensorToSelf {
    mtype _id;
}
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>86.</td>
<td>Time _ReadingTimer_body;</td>
</tr>
<tr>
<td>87.</td>
<td>Time _SleepTimer_body;</td>
</tr>
<tr>
<td>88.</td>
<td>Message _GetReading_body;</td>
</tr>
<tr>
<td>89.</td>
<td>Message _GotoSleep_body;</td>
</tr>
<tr>
<td>90.</td>
<td>Message _Wakeup_body;</td>
</tr>
<tr>
<td>91.</td>
<td>Message _DetermineEnergyLeft_body;</td>
</tr>
<tr>
<td>92.</td>
<td>Message _DetermineCH_body;</td>
</tr>
<tr>
<td>93.</td>
<td>}</td>
</tr>
<tr>
<td>94.</td>
<td></td>
</tr>
<tr>
<td>95.</td>
<td>typedef SensorToOtherSensor {</td>
</tr>
<tr>
<td>96.</td>
<td>mtype _id;</td>
</tr>
<tr>
<td>97.</td>
<td>Message _BecomeLeader_body;</td>
</tr>
<tr>
<td>98.</td>
<td>}</td>
</tr>
<tr>
<td>99.</td>
<td></td>
</tr>
<tr>
<td>100.</td>
<td>typedef SensorToClusterHead {</td>
</tr>
<tr>
<td>101.</td>
<td>mtype _id;</td>
</tr>
<tr>
<td>102.</td>
<td>Message _AcknowledgementSchedule_body;</td>
</tr>
<tr>
<td>103.</td>
<td>Reading _SensedData_body;</td>
</tr>
<tr>
<td>104.</td>
<td>Message _JoinRequest_body;</td>
</tr>
<tr>
<td>105.</td>
<td>Message _SendsCandidature_body;</td>
</tr>
<tr>
<td>106.</td>
<td>}</td>
</tr>
<tr>
<td>107.</td>
<td></td>
</tr>
<tr>
<td>108.</td>
<td>typedef ClusterHeadToSensor {</td>
</tr>
<tr>
<td>109.</td>
<td>mtype _id;</td>
</tr>
<tr>
<td>110.</td>
<td>Message _JoinAdvertisement_body;</td>
</tr>
<tr>
<td>111.</td>
<td>Schedule _SensorSchedule_body;</td>
</tr>
<tr>
<td>112.</td>
<td>Message _SubmitCandidature_body;</td>
</tr>
<tr>
<td>113.</td>
<td>UpdateParameterMessage _UpdateParameter_body;</td>
</tr>
<tr>
<td>114.</td>
<td>Message _GotoSleep_body;</td>
</tr>
<tr>
<td>115.</td>
<td>Message _WakeupMessage_body;</td>
</tr>
<tr>
<td>116.</td>
<td>}</td>
</tr>
<tr>
<td>117.</td>
<td></td>
</tr>
<tr>
<td>118.</td>
<td>typedef CHtoCH {</td>
</tr>
<tr>
<td>119.</td>
<td>mtype _id;</td>
</tr>
<tr>
<td>120.</td>
<td>Message _EnergyLeft_body;</td>
</tr>
<tr>
<td>121.</td>
<td>Time _StartJoinRequest_body;</td>
</tr>
<tr>
<td>122.</td>
<td>Time _CheckEnergy_body;</td>
</tr>
<tr>
<td>123.</td>
<td>Time _CheckARSTimer_body;</td>
</tr>
<tr>
<td>124.</td>
<td>Message _DetermineNewLeader_body;</td>
</tr>
<tr>
<td>125.</td>
<td>}</td>
</tr>
<tr>
<td>126.</td>
<td></td>
</tr>
<tr>
<td>127.</td>
<td>typedef ClusterHeadToARS {</td>
</tr>
<tr>
<td>128.</td>
<td>mtype _id;</td>
</tr>
<tr>
<td>129.</td>
<td>Message _ProbeARS_body;</td>
</tr>
<tr>
<td>130.</td>
<td>Message _JoinRequestARS_body;</td>
</tr>
</tbody>
</table>
typedef ARSToClusterHead {
    mtype_id;
    Message_ConfirmJoinRequest_body;
    Schedule_CHSchedule_body;
    UpdateParameterMessage_UpdateParameter_body;
    Message_RelinquishControlMessage_body;
    Message_RejoinARSMessage_body;
};

type ARStoSelf {
    mtype_id;
    Time_ActionTimer_body;
    Time_StartBSProbe_body;
    Message_StartAction_body;
};

type ARStoOtherARS {
    mtype_id;
    Query_RequestResultInfo_body;
    ResultSet_SendResultInfo_body;
};

type ARSToBaseStation {
    mtype_id;
    Message_SendStatus_body;
    Message_JoinRequestBaseStation_body;
    Reading_SendData_body;
    Query_QueryDataMessage_body;
    Message_AcknowledgementSchedule_body;
};

type BaseStationToARS {
    mtype_id;
    ResultSet_QueryResult_body;
    UpdateParameterMessage_UpdateParameter_body;
    Message_ConfirmJoinRequest_body;
    Schedule_ARSSchedule_body;
    Message_RelinquishControlMessage_body;
    Message_RejoinBSMessage_body;
    Message_GiveStatusInfo_body;
}
typedef BaseStationToBaseStation {
    mtype _id;
    Time _PurgeDataTimer_body;
    Time _StatusInfoTimer_body;
    Time _StartSinkProbe_body;
    Time _StartJoinRequest_body;
    Message _PurgeData_body;
};

typedef BaseStationToSink {
    mtype _id;
    Message _SendStatus_body;
    Message _JoinRequestSink_body;
    Message _AcknowledgementSchedule_body;
    Message _ProbeSink_body;
};

typedef SinkToBaseStation {
    mtype _id;
    UpdateParameterMessage _UpdateParameter_body;
    Schedule _CellSchedule_body;
    Message _ConfirmJoinRequest_body;
    Message _ReplaceComponent_body;
};

typedef SinkToSink {
    mtype _id;
    Time _CheckPerformanceTimer_body;
    Time _ReplaceComponentTimer_body;
};
REFERENCES


[16] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, Scott Shenker, “Data-centric storage in sensornets with GHT, a


doi: http://doi.acm.org/10.1145/872757.872817


doi: http://doi.acm.org/10.1145/844128.844142


doi: http://dx.doi.org/10.1504/IJAHUC.2008.017000


doi: http://dx.doi.org/10.1016/j.adhoc.2004.04.003


162


sensor networks,” Computer Communications [J], vol. 30, Issue 10, pp. 2274-2283,
July 2007. ISSN: 0140-3664. doi: 10.1016/j.comcom.2007.06.002

In the Proceedings of the 6th Annual ACM/IEEE International Conference on

978-0-7923-9697-0. doi: 10.1007/978-0-585-29603-6_5

[34] Perkins C and Royer E, “Ad Hoc On-Demand Distance Vector (AODV) Routing,”
doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.2139

Routing Protocol for Wireless Sensor Networks,” International Conference on
Wireless Communications, Networking and Mobile Computing (WiCom),
10.1109/WICOM.2007.630

Networks,” in the Proceedings of the 14th IEEE International Workshop on Quality

http://cast.cse.ohio-state.edu/exscal/.”

[38] Chen M, Leung V and Mao S et al., “Directional geographical routing for real-time
video communications in wireless sensor networks”, Computer Communication[J],

doi: http://doi.ieeecomputersociety.org/10.1109/MOBHOC.2007.4428692


doi: http://doi.ieeecomputersociety.org/10.1109/ICDCSW.2003.1203636


doi: http://dx.doi.org/10.1016/j.adhoc.2003.09.010


doi: http://dx.doi.org/10.1007/s10586-005-6183-7


doi: http://dx.doi.org/10.1016/j.comcom.2007.06.002


doi: 10.1109/COMSWA.2007.382429.


doi: 10.1109/WOCN.2007.4284155.


doi: 10.1109/IWADS.2000.880881


169


doi: http://doi.ieeecomputersociety.org/10.1109/RELDI.2000.885410


http://stwww.weizmann.ac.il/g-cs/benari/jSPIN/


doi: http://doi.acm.org/10.1145/958491.958508


173


PUBLICATIONS

International Journals

   WebSite: http://www.scipub.org/fulltext/jcs/jcs5121063-1070.pdf


   doi: 10.4236/wsn.2010.21008,

   doi : http://indjst.org/archive/vol.2.issue.11-12/nov09sanjeev-11.pdf

International Conferences

doi : 10.1109/IADCC.2010.5422928


Submitted Manuscript


Manuscript under Preparation
