



**ENERGY EFFICIENT CLUSTER BASED
DATA COLLECTION FOR WIRELESS
SENSOR NETWORK**



A PROJECT REPORT

Submitted by

VIJAYALAKSHMI M

*in partial fulfillment for the requirement of award of the degree
of*

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

**KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE 641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

**ENERGY EFFICIENT CLUSTER
BASED DATA COLLECTION FOR
WIRELESS SENSOR NETWORK**

A PROJECT REPORT

Submitted by

VIJAYALAKSHMI M

*in partial fulfillment for the requirement of award of the degree
of*

MASTER OF ENGINEERING



FACULTY OF COMPUTER SCIENCE AND ENGINEERING

**KUMARAGURU COLLEGE OF
TECHNOLOGY COIMBATORE 641 049**

**(An Autonomous Institution Affiliated to Anna University,
Chennai)**

APRIL 2013

iii

BONAFIDE CERTIFICATE

Certified that this project work titled “ **ENERGY EFFICIENT CLUSTER BASED DATA COLLECTION FOR WIRELESS SENSOR NETWORK**” is the bonafide work of Ms. M.VIJAYALAKSHMI (1120108023), who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other students.

Prof.N.Jayapathi M.Tech.,

PROFESSOR AND HEAD

Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore- 641 049

Dr.V.Vanitha M.E., Ph.D

SUPERVISOR

Senior Associate Professor

Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore- 641 049

Submitted for the Project Viva-Voce examination held on _____.

Internal Examiner

External Examiner

iv

ABSTRACT

For many applications in Wireless Sensor Networks (WSNs), users may want to continuously extract data from the networks for analysis later. However, accurate data extraction is difficult. It is often too costly to obtain all sensor readings, as well as not necessary in the sense that the readings themselves only represent samples of the true state of the world. Clustering and prediction techniques, which exploit spatial and temporal correlation among the sensor data provide opportunities for reducing the energy consumption of continuous sensor data collection.

Integrating clustering and prediction techniques makes it essential to design a new data collection scheme, so as to achieve network energy efficiency and stability. Proposed framework is clustering based. A cluster head represents all sensor nodes in the cluster and collects data values from them. An energy-efficient framework is proposed for clustering-based data collection in wireless sensor networks by integrating adaptively enabling/disabling prediction scheme.

To realize prediction techniques efficiently in WSNs, adaptive scheme to control prediction is to be used in the proposed framework, analyze the performance trade off between reducing communication cost and limiting prediction cost, and design algorithms to exploit the benefit of adaptive scheme to enable/disable prediction operations.

Sleep/awake scheduling can be applied, which takes proposed framework approach to design a practical algorithm for data aggregation. It avoids the need for rampant node-to-node propagation of aggregates, but rather it uses faster and more efficient cluster-to-cluster propagation.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for enabling me to complete this project. I express my profound gratitude to **Padmabhusan Arutselvar Dr.N.Mahalingam B.Sc.,F.I.E., Chairman, Dr.B.K.Krishnaraj Vanavarayar B.com., B.L., Co-Chairman, Mr.M.Balasubramaniam M.com.,M.B.A., Corresspondent, Mr.K.Shankar Vanavarayar M.B.A., PGDIEM, Joint Corresspondent and Dr.S.Ramachandran Ph.D., Principal** for providing the necessary facilities to complete my project.

I take this opportunity to thank **Prof.N.Jayapathi M.Tech.**, Head Of the Department, Department of Computer Science and Engineering, for his support and motivation.

I thank all project committee members for their comments and advice during the reviews. Special thanks to my Project Coordinator and Guide **Dr.V.Vanitha M.E., Ph.D.**, Senior Associate Professor, Department of Computer science and Engineering, for arranging brain storming project review sessions. I am grateful for her support, encouragement and ideas. I would like to convey my honest thanks to all Teaching and Non Teaching staff members of the department for their support.

I dedicate this project work to my Parents for no reasons but feeling from bottom of my heart that without their love this work wouldn't be possible.

-M.VIJAYALAKSHMI

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
	1.1 OVERVIEW OF WIRELESS SENSOR NETWORKS	1
	1.2 APPLICATIONS OF WIRELESS SENSOR NETWORKS	4
	1.3 CHARACTERISTICS	5
	1.4 WIRELESS SENSOR NETWORK ARCHITECTURE	5
	1.4.1 Layered Architecture	5
	1.4.2 Clustered Architecture	7
	1.5 DESIGN CHALLENGES OF WSNs	7
	1.6 SIMULATION OF WSNs	9
	1.7 LITERATURE SURVEY	10
	1.7.1 CLUSTERING APPROACHES	10
	1.7.1.1 Low-Energy Adaptive Clustering Hierarchy	10
	1.7.1.2 Hierarchical Cluster-based Routing in WSNs	12
	1.7.1.3 HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-Hoc WSN	14
	1.7.1.4 A Distributed Efficient Clustering Approach for Ad hoc and Sensor Networks (DECA)	16
	1.7.1.5 ASAP: An Adaptive Sampling Approach to Data	18

	Collection in Sensor Networks	
	1.7.2 TREE BASED APPROACHES	19
	1.7.2.1 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)	20
	1.7.2.2 Computing Localized Power Efficient Data Aggregation Trees for Sensor Networks	21
	1.7.3 SLEEP/WAKE SCHEDULING	22
	1.7.3.1 Optimal Sleep/Wake Scheduling for Time-Synchronized WSN with QOS Guarantee	22
	1.7.3.2 Energy Efficient Sleep/Wake Scheduling for Multi-Hop Sensor Networks: Non-Convexity and Approximation Algorithm	26
	1.7.4 COMPARATIVE STUDY OF ENERGY EFFICIENT CLUSTERING PROTOCOLS	27
2	IMPLEMENTATION OF ENERGY EFFICIENT CLUSTER BASED DATA COLLECTION FOR WSN	29
	2.1 PROBLEM DEFINITION	29
	2.2 PROJECT DESCRIPTION	29
	2.2.1 System Architecture and Design Issues	30
	2.2.2 Clustering Approach	32
	2.2.3 Adaptive Scheme to Enable/Disable Prediction Operation	34
	2.2.4 Sleep / Awake Scheduling	36
3	RESULT AND ANALYSIS	39
	3.1 SYSTEM SPECIFICATION	39
	3.1.1 HARDWARE REQUIREMENTS	39
	3.1.2 SOFTWARE REQUIREMENTS	39
	3.1.3 SOFTWARE DESCRIPTION	39
	3.2 IMPLEMENTAION	40

	3.2.1 Simulation Environment	40
	3.2.2 Snap Shots	40
	3.3 ANALYSIS	43
	3.3.1 Cluster Model	43
	3.3.2 Average Energy Consumption	43
	3.3.3 Throughput	43
	3.3.4 Performance Evaluation	43
	3.4 CONCLUSION AND FUTURE WORK	47
	APPENDIX	48
	Sample Source Code	48
	REFERENCES	63
	LIST OF PUBLICATIONS	65

LIST OF TABLE

TABLE NO.	TITLE	PAGE NO.
1.1	Comparison of Energy Efficient Clustering Protocols	28

LIST OF FIGURES

FIGURE NO.	CAPTION	PAGE NO.
1.1	WSN Architecture	1
1.2	Components of Sensor Node	2
1.3	Layered Architecture	6
1.4	Clustered Architecture	7
1.5	Distributed cluster formation algorithm for LEACH	11
1.6	Cluster-based hierarchical model	13
1.7	Chain construction using greedy algorithm	20
1.8	Three level cluster hierarchy	27
3.1	Snapshot1 of Proposed Framework	41
3.2	Snapshot2 of Proposed Framework	42
3.3	Snapshot3 of Proposed Framework	42
3.4	Average Energy Consumption for proposed scheme	44
3.5	Average Throughput with and without proposed schemes	45
3.6	Average Energy Consumption in proposed scheme while CH role rotated, with and without proposed scheme	45

LIST OF ABBREVIATIONS

BS	Base Station
CH	Cluster Head
QOS	Quality Of Service
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF WIRELESS SENSOR NETWORKS

Wireless Sensor Networks (WSNs) comprise of a higher number of nodes (in the thousands and more) scattered over some region. Sensor nodes are typically less mobile, and more densely deployed than mobile ad hoc networks (MANETs). The sensor nodes gather data from the environment and can perform various kinds of activities such as collaborative processing of the sensor data, and performing some synchronized actions based on the gathered sensor data. This WSN architecture is depicted in the Fig. 1.1.

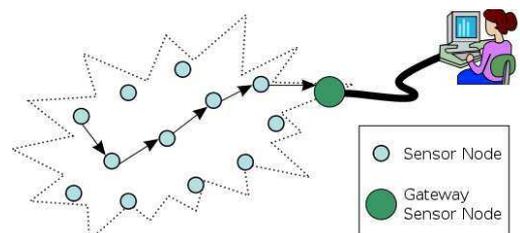


Fig. 1.1 WSN Architecture

It is not unreasonable to expect that in 10-15 years that the world will be covered with wireless sensor networks with access to them via the Internet. This can be considered as the Internet becoming a physical network.

1.1.1 Sensor Node

The Wireless Sensor Network (WSN) is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A wireless sensor node is composed of four basic components: a sensing unit, a processing unit (microcontroller), a transceiver unit and a power unit as shown in Fig. 1.2.

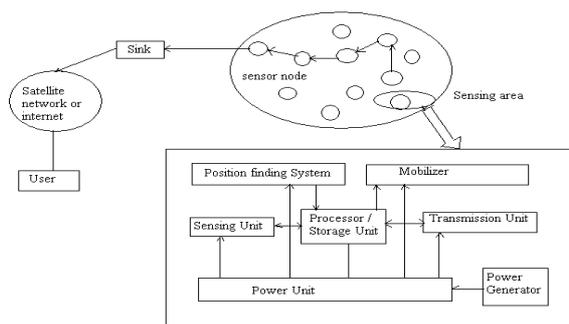


Fig. 1.2 Components of Sensor Node

In addition to the above units, a wireless sensor node may include a number of application-specific components, for example a location detection system or mobilizer; for this reason, many commercial sensor node products include expansion slots and support serial wired communication.

Sensing Unit

The main functionality of the sensing unit is to sense or measure physical data from the target area. The analog voltage or signal is generated by the sensor corresponding to the observed phenomenon. The continual waveform is digitized by an Analog-to-Digital Converter (ADC) and then delivered to the processing unit for further analysis.

Processing Unit

The processing unit which is generally associated with a small storage unit manages the procedures that make the sensor nodes collaborate with the other nodes to carry out the assigned sensing tasks.

Transceiver

There are three deploying communication schemes in sensors including optical communication (laser), infrared, and Radio-Frequency (RF). Laser consumes less energy than radio and provides high security, but requires line of sight and is sensitive to atmospheric conditions. Infrared, like laser, needs no antenna but is limited in its broadcasting capacity. RF is the most easy to use but requires antenna. Various energy consumption reduction strategies have been developed such as modulation, filtering, and demodulation. Amplitude and frequency modulation are standard mechanisms.

Power Unit

Every sensor node is equipped with a battery that supplies power to remain in active mode. Power consumption is a major weakness of sensor networks. Any energy preservation schemes can help to extend sensor's lifetime. Batteries used in sensors can be categorized into two groups; rechargeable and non-rechargeable. Often in harsh environments, it is impossible to recharge or change a battery.

APPLICATIONS OF WSNs

Area monitoring

Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors to detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines. Similarly, WSNs are used to detect the presence of vehicles ranging from motorcycles to train cars.

Environmental/Earth monitoring

The term Environmental Sensor Networks, has evolved to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, forests, etc.

Air quality monitoring

Real time monitoring of dangerous/pollutant gases are particularly interesting and utmost important in hazardous areas, as the conditions can change dramatically very quickly, with serious consequences.

Natural disaster prevention

Wireless sensor networks can effectively act to prevent the consequences of natural disasters, like floods, fire in forests. Wireless nodes have successfully been deployed in rivers where changes of the water levels have to be monitored in real time.

Structural monitoring

Wireless sensors can be used to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc. enabling Engineering practices to monitor assets remotely without the need for costly site visits. It is also far more accurate than any visual inspection that would be carried out.

1.3 CHARACTERISTICS

The main characteristics of a WSN include Power consumption constrains for nodes using batteries or energy harvesting, and also it includes the ability to cope with node failures, mobility of nodes, communication failures, heterogeneity of nodes, scalability to large scale of deployment, ability to withstand harsh environmental conditions, ease of use, power efficiency to prolong network lifetime.

1.4 WIRELESS SENSOR NETWORKS ARCHITECTURES

1.4.1 Layered Architecture

In this type of architecture there is a single powerful base station (BS) and layers of sensor nodes are formed around BS based on their hop count distance

to reach BS. Therefore, in general layer i denote all nodes that are i -hop away from BS. Layered architecture is depicted in Fig. 1.3.

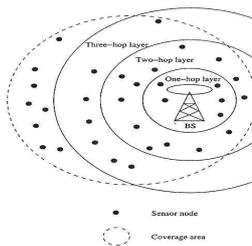


Fig. 1.3 Layered Architecture

Unified Network Protocol Framework (UNPF)

It is a type of layered architecture with a set of protocols that integrates the following operations:

Network Initialization & Maintenance Protocol

At network initialization, BS broadcasts its ID using CDMA common control channel (BS reaches all nodes in one hop). Nodes record base station ID & send beacon signal with their own IDs at their low default power levels. The BS broadcasts a control packet with all layer one node IDs. All nodes send a beacon signal again. The layer one nodes record the IDs they hear from layer two and inform the BS about layer two. The BS broadcasts the layer two nodes IDs. For maintenance, periodic beaconing updates are required. MAC protocol uses a Time Division CDMA (TCDMA) protocol for spatial bandwidth reuse and ensures a scheduling scheme for fair access.

1.4.2 Clustered Architecture

In this type of architecture sensor nodes are organized into clusters and each cluster is governed by a cluster-head. Only cluster heads send messages to a BS. This architecture is suitable for data fusion and is self-organizing in nature. This is depicted in Fig. 1.4.

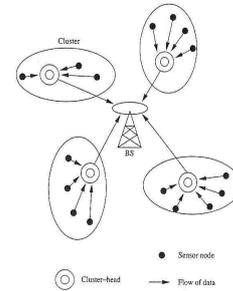


Fig. 1.4 Clustered Architecture

1.5 DESIGN CHALLENGES OF WSN

On the downside, the large and growing number of networked sensors present a number of unique system design challenges as listed below (Chu 2006).

1. Scalable and flexible architecture

The network must preserve its stability. Introducing more nodes into the network means that additional communication messages will be exchanged, so that these nodes are integrated into the existing network.

2. Sensors are power constrained.

Sensors are usually heavily resource-constrained (especially on power), irreplaceable, and become unusable after failure or energy depletion. It is thus crucial to devise novel energy-efficient solutions for topology organization and routing that are scalable, efficient and energy conserving in order to increase the overall network longevity (Heinzelman 2002).

3. Sensor networks must deal with high system dynamics

Sensor devices and sensor networks experience a wide range of dynamics, including spatial and temporal change trends in the sensed values that contribute to the environmental dynamics, changes in the user demands that contribute to the task dynamics as to what is being sensed and what is considered interesting changes, and changes in the energy levels, location, or connectivity of the sensor nodes that contribute to the network.

4. Fault tolerance and adaptability

Fault tolerance means to maintain sensor network functionalities without any interruption due to failure of sensor node because in sensor network every node have limited power of energy so the failure of single node doesn't affect the overall task of the sensor network.

6. Short Range Transmission

The WSNs should consider the short transmission range in order to reduce the possibility of being eavesdropped.

7. Limited computational power and memory size

It is another factor that affects WSN in the sense that each node stores the data individually and sometime more than one node stored same data and transferred to the base station which waste the power and storing capacity of nodes. so the effective routing schemes and protocols must be developed to minimize the redundancy in the network

8. Security

Security is very important parameter in sensor network since sensor networks are data centric so there is no particular id associated with sensor nodes and attacker can easily inserted himself into the network and stole the important data by becoming the part of network without the knowledge of sensor nodes of the network. So it is difficult to identify whether the information is authenticated or not.

These properties of WSN impose unique challenges for development of communication protocols in such architecture. The intrinsic properties of individual sensor nodes, pose additional challenges to the communication in terms of energy consumption.

1.6 SIMULATION OF WSNs

In general, there are two ways to develop simulations of WSNs. First option is to use a custom platform to develop the simulation and the second option is to develop one's own simulation. As such, at present Agent-based Modeling and Simulation is the only paradigm which allows the simulation of even complex behavior in the environments of Wireless sensors (such as flocking). Network Simulators like OPNET, NetSim and NS2 can be used to simulate Wireless Sensor Network.

1.7 LITERATURE SURVEY

In this section, papers related to energy efficient clustering protocols and sleep/wake scheduling for WSNs are discussed.

1.7.1 CLUSTERING APPROACHES

Clustering is an efficient mechanism for improving the energy efficiency. In clustering schemes, sensor nodes are dividing into a number of small clusters. Each cluster has a coordinator/CH, and a number of cluster members. Cluster members can transmit data to their own CH directly, while CHs collect the data and send them to the sink node. Some of the famous clustering schemes are analyzed under this chapter.

1.7.1.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

It is a self-organizing and adaptive clustering protocol which evenly distributes the energy expenditure among the sensors. It performs data aggregation where cluster heads act as aggregation points. There are two main phases in this architecture (Gedik 2007).

Setup phase

This phase organizing the clusters. Each sensor chooses a random number m between 0 and 1. If $m < T(n)$ for node n , the node becomes a cluster-head where

$$T(n) = \begin{cases} \frac{P}{1 - P[r \bmod (1/P)]} & \text{if } n \in G \\ 0 & \text{otherwise,} \end{cases}$$

P : the desired percentage of cluster heads

r : the round number

G : the set of nodes that have not been cluster heads during the last $1/P$ rounds.

A cluster head advertises its neighbors using a CSMA MAC. Surrounding nodes decide which cluster to join based on the signal strength of these messages. Cluster heads assign a TDMA schedule for their members.

Steady-state phase

The steady-state phase deals with the actual data transfers to the BS. All source nodes send their data to their CHs. CHs perform data aggregation/fusion through local transmission and send them back to BS using a single direct transmission. After a certain period of time, CHs are selected again through the set-up phase. This processing is pictorially represented in Fig. 1.5.

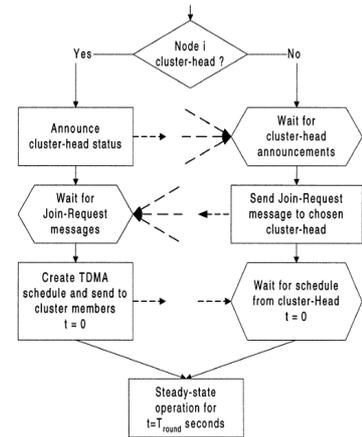


Fig. 1.5 Flowchart of the distributed cluster formation algorithm for LEACH.

Pros

LEACH evenly distributes the energy load among the sensors in the network and thereby it prevents energy drain for the same sensor nodes which has been elected as the cluster head. It is 4 to 8 times effective over direct communication / MTE approach. It provides opportunity to implement any aggregation function at the cluster heads. Cluster collisions were solved by unique TDMA/CDMA codes for each cluster.

Cons

There is no discussion on optimal CH selection and it has highly dynamic environments. Because of the random selection of CH, in the worst case the CH nodes may not be evenly distributed among the nodes. It also requires continuous updates and mobility also takes major impact.

1.7.1.2 Hierarchical Cluster-based Routing in Wireless Sensor Networks

It is an extension of LEACH. As shown in Fig. 1.6, a hierarchical approach breaks the network into clustered layers. Nodes are grouped into clusters with a cluster head that has the responsibility of routing from the cluster to the other cluster heads or BS.

It performs two-layer routing. Data travel 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 larger distances. This moves data faster to the BS. In the cluster-based hierarchical model, data is first aggregated in the cluster then sent to a higher-level cluster-head. Higher energy nodes can be used to process and send the

information. Low energy nodes can be used to perform the sensing in the proximity of the target.

Merits

As it moves from a lower level to a higher one, it travels greater distances, thus reducing the travel time and latency. This model is better than the one hop or multi-hop model. It greatly contributes to overall system scalability, lifetime, and energy efficiency. In cluster-based model, only cluster-heads performs data aggregation whereas in multi-hop model, every intermediate node performs data aggregation. As a result, the cluster-based model is more suitable for time-critical applications than the multi-hop model.

Demerit

As the distance between clustering level increases, the energy spent is proportional to the square of the distance. This increases energy expenditure.

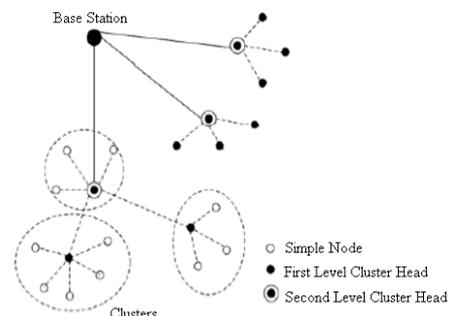


Fig. 1.6 Cluster-based hierarchical model

1.7.1.3 HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-Hoc Sensor Networks

It elects CHs based on residual energy and node degree or density of nodes as a metric for cluster selection to achieve power balancing, which is a rational improvement compared with LEACH. In HEED, the proposed algorithm periodically selects CHs according to a combination of two clustering parameters. The primary parameter is their residual energy of each sensor node and the secondary parameter is the intra-cluster communication cost as a function of cluster density. The primary parameter is used to probabilistically select an initial set of CHs while the secondary parameter is used for breaking ties (Baranidharan 2010).

Protocol Operation

In HEED, the clustering process at each sensor node requires several rounds. Every round is long enough to receive messages from any neighbor within the cluster range. Each round involves the following phases. At Initialization phase, Each sensor node computes its probability to become CH using the formula $CH_{prob} = C_{prob} \cdot (E_{residual} / E_{max})^I$. This processing is continued until it find out the node with higher residual energy. Then the node with higher residual energy is finalized as CH.

The parameter C_{prob} is only used to limit the initial CH announcements and has no direct impact on the final cluster structure. In HEED, each sensor node sets the probability CH_{prob} of becoming a CH as follows. $E_{residual}$ is the estimated current residual energy in this sensor node and E_{max} is the maximum energy corresponding to a fully charged battery, which is typically identical for homogeneous sensor nodes. The CH_{prob} value must be greater than a minimum

threshold p_{min} . A CH is either a tentative CH, if its $CH_{prob} < 1$, or final CH, if its CH_{prob} has reached 1.

During each round of HEED, every sensor node that never heard from a CH elects itself to become a CH with probability CH_{prob} . The newly selected CHs are added to the current set of CHs. If a sensor node is selected to become a CH, it broadcasts an announcement message as a tentative CH or a final CH. A sensor node hearing the CH list selects the CH with the lowest cost from this set of CHs. Every node then doubles its CH_{prob} and goes to the next step.

Advantages

- Rational improvement in energy efficiency compared with LEACH.
- Terminating the clustering process within a constant number of iterations.
- Minimizing control overhead.
- HEED does not require special node capabilities, such as location-awareness and HEED does not make assumptions about node distribution.
- The nodes also automatically update their neighbor sets in multi-hop networks by periodically sending and receiving messages.
- It operates correctly even when nodes are not synchronized.
- Producing well-distributed CHs and compact clusters

Disadvantages

- The random selection of the CHs may cause higher communication overhead for the ordinary member nodes in communicating with their corresponding CH, the CHs in establishing the communication among them, or between a CH and a base station.

- The periodic cluster head rotation or election needs extra energy to rebuild clusters.

1.7.1.4 A Distributed Efficient Clustering Approach for Ad hoc and Sensor Networks (DECA)

The DECA algorithm structure is somewhat similar to that presented in Adaptive clustering in mobile wireless networks and the HEED protocol in that each node broadcasts its decision as the cluster head in the neighbourhood based on some local information and score function. The difference between DECA and these two protocols lies in when and how the nodes make such decisions and how the score gets computed (Meka 2006).

DECA Operation

Each node periodically transmits a Hello message to identify itself, and based on such Hello messages, each node maintains a neighbour list. It define the score function for each node as $score = w E + w C + w I$, where E stands for the node residual energy, C stands for the node connectivity, I stands for the node identifier, w stands for weight and the weights follow $\sum_{i=1}^3 w_i = 1$.

The computed score is then used to compute the delay for this node to announce itself as the CH. The higher the score, the sooner the node will transmit. The computed delay is normalized between 0 and a certain upper bound, which is a key parameter that needs to be carefully selected. After the clustering starts, the procedure will terminate after time T_{stop} , another key parameter whose selection needs to take the node computation capability and mobility into consideration.

Upon receiving clustering messages, a node first checks whether the node ID and the cluster ID embedded in the received message are the same. Same node and cluster ID means that the message has been transmitted from a cluster head. Further, if the receiving node does not belong to any cluster, and the received score is better than its own, the node will mark down the advertised cluster and wait until its scheduled time to send its message.

If the receiving node currently belongs to some cluster, and the received score is better than its own score, two cases are further considered. First, if the current node receiving a better-scored message is not a cluster head itself, as an ordinary node, it can immediately mark down the best cluster so far and wait until its scheduled announcement. This node will stay in its committed cluster after its announcement.

On the other hand, if the current node is a CH itself, receiving a better scored message means that this node may need to switch to the better cluster. However, cautions need to be taken here before switching since the current node, as a CH, may already have other nodes affiliated with it. Therefore, inconsistencies can occur if it rushes to switch to another cluster.

When new nodes and already existing nodes from some other cluster forming a cluster with a new CH, the CHs ID, cluster ID and score value should be broadcasted. In the finalizing phase, where each node is forced to enter after stop T , each node checks to see if it needs to convert. Further, each node checks if it already belongs to a cluster and will initiate a new cluster with itself as the head if not so.

Theoretical Results

- Eventually DECA terminates.
- At the end of actual announcement phase, every node can determine its cluster and only one cluster.
- When clustering finishes, any two nodes in a cluster are at most two-hops away.
- In DECA, each node transmits only one message during the operation.
- The time complexity of DECA is $O(|V|)$.

Pros

DECA outperforms HEED with about twice the CH residual energy. It generates non-overlapping clusters with good clustering performance. Perhaps it consistently incurs fewer message transmissions than HEED. It provides the scalability and prolonged network lifetime and it is resilient to node mobility.

Cons

The performance gain of DECA over HEED decreases as the transmission range increases.

1.7.1.5 ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks

One of the most prominent and comprehensive ways of data collection in sensor networks is to periodically extract raw sensor readings. This way of data collection enables complex analysis of data, which may not be possible with in-network aggregation or query processing. However, this flexibility in data analysis comes at the cost of power consumption.

The main idea behind ASAP is to use a dynamically changing subset of the nodes as samplers such that the sensor readings of the sampler nodes are directly collected, whereas the values of the non sampler nodes are predicted through the use of probabilistic models that are locally and periodically constructed. It is effectively used to increase the network lifetime while keeping the quality of the collected data high in scenarios where either the spatial density of the network deployment is superfluous, which is relative to the required spatial resolution for data analysis, or certain amount of data quality can be traded off in order to decrease the power consumption of the network (Gedik 2007).

ASAP is a Hybrid Approach, in which Spatial / temporal correlations are summarized locally within network and Value prediction is performed centrally at BS.

Merits

- It increases the network lifetime and also keeps the quality of the collected data
- Results in energy consumption
- When compared to No-sampling approach, ASAP Provides
 - 25 to 45% savings when sampling fraction(σ) ranges from 0.5 to 0.1
 - 38 to 44% savings when node density increase from 5 to 15/unit circle

1.7.2 TREE BASED APPROACHES

Instead of having layered architecture, sometimes the nodes will transfer to and obtain data from its neighbors only and this type of architecture is called as tree based approaches.

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

The main idea in PEGASIS is for each node to receive from and transmit to close neighbors and take turns being the leader for transmission to the BS. This approach will distribute the energy load evenly among the sensor nodes in the network (McConnell 2005).

For gathering data from sensor nodes in each round, each node receives data from one neighbor, fuses the data with its own, and transmits to the other neighbor on the chain. The leader in each round of communication will be at a random position on the chain, which is important for nodes to die at random locations. The idea of nodes dying at random places is to make the sensor network robust to failures.

Each round of data collection can be initiated by BS with a beacon signal which will synchronize all sensor nodes. Since all nodes know their positions on the chain, a time slot approach can be used for transmitting data. In the i^{th} round of data collection, node $c(i-1)$ will be the leader. The end node c_0 will transmit its data to node c_1 in slot one, c_1 fuses and transmits data in slot two, and so on until the leader node is reached. This is illustrated in Fig. 1.7.

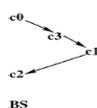


Fig. 1.7 Chain construction using greedy algorithm

H-PEGASIS (Hierarchical PEGASIS)

An extension to PEGASIS, called Hierarchical-PEGASIS was introduced with the objective of decreasing the delay incurred for packets during transmission to the BS. In H-PEGASIS, simultaneous transmissions of data messages are pursued and it proposes a solution to the data gathering problem by considering energy X delay metric.

Pros

Each node communicates only with a close neighbor and takes turns transmitting to the BS, thus reducing the amount of energy spent per round. It is approximately two times better than LEACH. Power draining is spread uniformly over all nodes and avoids collisions by using two approaches: 1. CDMA, 2. spatially separated nodes are allowed to transmit at the same time.

Cons

It assumes direct communication. But in practical, nodes use multi-hop communication to reach the BS. It assumes that all sensor nodes have the same level of energy and are likely to die at the same time. PEGASIS introduces excessive delay for distant nodes on the chain. The single leader can become a bottleneck.

1.7.2.2 Computing Localized Power Efficient Data Aggregation Trees for Sensor Networks

It combines the desired properties of minimum spanning tree and shortest path tree-based routing schemes and considers the remaining power levels

of nodes which increase network lifetime. It calculating both Source and Sink energy to find the best node for communication (Sanjay Waware 2012).

Pros

Its an adaptive approach and perfect aggregation is used (Power efficient data aggregation). It provides same performance of a centralized solution in terms of network lifetime.

Con

L-PEDAP Communication Time is more as compared with that of PEDAP. It calculating both Source and Sink energy to find the best node for communication.

1.7.3 SLEEP/WAKE SCHEDULING

Sleep/awake scheduling is one of the common mechanism which is used in WSNs in order to improve the energy consumption and thereby prolonging the network lifetime. The basic idea of Sleep/Wake scheduling is to put the radio to sleep during idle times, and wake it up right before message transmission/reception.

1.7.3.1 Optimal Sleep/Wake Scheduling for Time-Synchronized WSN with QoS Guarantee

It studies the sleep/wake scheduling for low-duty cycle sensor networks and show that its synchronization error is non-negligible, and using a conservative guard time is energy wasteful. Therefore, this proposed model formulate an optimization problem that aims to set the capture probability threshold for messages from each individual node such that the expected energy consumption is minimized,

and the collective Quality of Service (QoS) over the nodes is guaranteed (Wu 2006).

Sleep/Wake Scheduling for Sensor Networks

Existing sleep/wake scheduling schemes can be synchronization-based, where nodes synchronize each other to coordinate their wake up schedules, or asynchronous/random which do not involve explicit synchronization. For continuous monitoring systems, synchronization-based sleep/wake scheduling schemes are often used because the traffic pattern is periodic. Fine-grained synchronization is required between the sender and the receiver, so that they can wake up at the same time to communicate.

Prior work either assumes that the underlying synchronization protocol can provide nearly perfect (e.g., micro-second level) synchronization, or assumes an upper bound on the clock disagreement, and uses it as a guard time to compensate for the synchronization error.

Existing synchronization protocols like RBS or TPSN achieve micro-second level synchronization at the time instant 'immediately following' the exchange of synchronization messages. Due to estimation errors in the clock skew, the clocks will gradually drift as time progresses, until the next exchange of synchronization messages. To see how significant the clock disagreement can be, consider two nodes that have agreed to rendezvous on the radio channel once every 100 seconds to exchange a 20-byte message. Using a 19.2 kbps radio such as RF Monolithics , 20 bytes can be transmitted in about 8 ms. The radio must be awakened early to account for clock disagreement. Let the estimation error of the clock skew be 10 parts-per-million (ppm)1, i.e., the clocks of the two nodes drift away from each other 10 μ s each second. After 100 seconds, the clocks will drift by

$10 \mu\text{s} \times 100 = 1 \text{ ms}$, which is non-negligible compared to the actual message transmission time.

The author (Wu 2006) considered the effect of synchronization error in the design of a polling-based MAC protocol called Scheduled Channel Polling (SCP). In SCP, the receivers periodically poll the channel for network activity, and the sender uses a preamble to wake up the receiver before sending the actual message. To accommodate the clock disagreement they extend the preamble by a guard time, which is equal to the product of the maximum clock skew and the time elapsed since last synchronization.

Assumptions

- Intra-cluster communication is focused. Each transmission interval is further divided into two subintervals. One is for intra-cluster communications, and the other is for inter-cluster communications when the CH is always active. In the remainder of the paper, only intra cluster communications is focused and further assume that neighboring clusters use orthogonal frequency channels and do not interfere with each other.
- Because of imprecision in the manufacturing process and aging effects, the frequency of a crystal oscillator may be different from its desirable value. The maximum clock skew is no larger than 100 ppm. The frequency is also affected by environmental factors including variations in temperature, pressure, voltage, radiation, and magnetic fields.
- For the transmitter circuit, it is assumed that the sender can precisely control when the message is sent out onto the channel using its own clock. For the receiver circuit, it is assumed that if there is an incoming message, the signal can be detected immediately. Once the receiver circuit detects an incoming

message, it can let the processor know, so that the processor will keep the radio active until the reception is completed.

- With recent advances in hardware technology, the transition time between sleep and wake states can be reduced to a few clock cycles. Thus, it is consider that the transition time is negligible.
- It is assumed that the separation between transmissions from different members is large enough so that the collision probability between transmissions from different members is negligible.
- Measurements show that among all the sensor components, the radio consumes the most significant amount of energy. Therefore, only energy consumption of the radio is taken into account.
- The propagation delay is below 1 μ s. Hence, it is negligible and assumed to be 0

Advantages

- By considering the worst case clock disagreement as the guard time can compensate for the synchronization error.
- Energy efficiency can be further improved by exploiting the nondeterministic nature of the synchronization error.
- It is conclude that the design of any sleep/wake scheduling algorithm must take into account the impact of synchronization error, and the optimal sleep/wake scheduling scheme with the consideration of the synchronization error. It obtains a solution with energy consumption that is provably at most 37% larger than the optimal solution.

Drawbacks

The computational complexity of the scheduling algorithm is very low. How to assign the capture thresholds under complex correlation patterns is a

challenging problem for future work. The single hop intra- cluster communication scenario is only focused.

1.7.3.2 Energy Efficient Sleep/Wake Scheduling for Multi-Hop Sensor Networks: Non-Convexity and Approximation Algorithm

In this work, multi-hop communications and hierarchically clustered network are considered. The previous assumption that the capture probability threshold is ‘already given’ is removed, and how to decide the per-hop capture probability thresholds to meet the QoS requirement of the application is studied. In many sensor network applications, the nodes collect sensing data and report to the BS. An optimization problem is formulated which aims to set the capture probability threshold at each hop such that the network lifetime is maximized, while a minimum fraction of data is guaranteed to be delivered to the BS. The problem turns out to be non-convex and hard to solve exactly. It obtains a 0.73 approximation algorithm (Wu 2006).

Assumptions

The following assumptions are taken into account.

- It is assumed that neighboring clusters use orthogonal frequency bands and do not interfere with each other. Because the data rate of sensor networks is usually low, typically around 10-40 kbps. A node that is both a CH and cluster member needs to communicate with its members and with its CH, e.g., in Fig. 1.8 node C has to schedule carefully to participate in each cluster. This can be achieved in the following manner. The BS first decides the schedule of the synchronization interval and the transmission schedule for its members (A and B in Fig.1.8), then broadcasts this information to the members. A and B, upon hearing the broadcast, will reserve the relevant times for communicating with

the root. Then, A and B schedule the synchronization and transmissions for their members at different times.

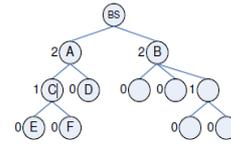


Fig. 1.8 Three level cluster hierarchy

- This model assumes the same compression ratio for messages from different nodes. It can be easily extended to account for different compression ratios.
- System measurements have shown that non-determinism at the sender is negligible compared to non-determinism at the receiver. For the receiver, it is assumed that if there is an incoming message, it can immediately detect the radio signal. Once the receiver detects an incoming message, it will stay active until the reception is completed.
- It is assumed that the separation between transmissions from different members, for a cluster with M members is large enough so that the collision probability for transmissions from different members is negligible. For low duty cycle networks, the message size is usually not large; hence the transmission time is much smaller than this separation.
- The propagation delay is negligible and assumes it to be zero for simplicity.

1.7.4 COMPARATIVE STUDY OF ENERGY EFFICIENT CLUSTERING PROTOCOLS

Table 1.1 gives comparison of various power efficient routing protocols discussed for wireless sensor networks (Baranidharan 2010).

- PEGASIS increases network lifetime two-fold compared to the LEACH .

- The HEED clustering improves network lifetime over LEACH clustering because LEACH randomly selects CHs, which may result in faster death of some nodes.
- The final CHs selected in HEED are well distributed across the network and the communication cost is minimized as compared to other routing protocols.
- DECA has twice the energy efficiency than HEED in terms of CH residual energy.
- HCR and ASAP too have considerable energy efficiency.

Table 1.1 Comparison of Energy Efficient Clustering Protocols

Protocols Parameters	LEACH	HCR	HEED	DECA	ASAP	PEGASIS
Routing	Cluster Based	Cluster Based	Cluster Based	Cluster Based	Hybrid	Chain Based
Mobility	Fixed BS	Fixed BS	Resilient to node mobility	Resilient to node mobility	Stationary	Fixed BS
Cluster Stability	Moderate	Good	High	Good	Dynamic	N/A
Energy Efficiency	No	Yes	Yes	Twice the HEED in terms of CH residual energy	Yes	2 times more than LEACH
Latency	Acceptable	Reduced than LEACH	Acceptable	Acceptable	Acceptable	Higher
Distributed CHs	Moderate	Moderate	Good	Good	-	N/A
Multihop	No	Yes	Yes	Yes	Yes	No

CHAPTER 2

IMPLEMENTATION OF ENERGY EFFICIENT CLUSTER BASED DATA COLLECTION FOR WSN

2.1 PROBLEM DEFINITION

Wireless sensor networks (WSNs) are composed of set of tiny sensor nodes, which can effectively monitor their surrounding environment. Due to the wide potential applications in battlefield surveillance, environmental monitoring, healthcare, weather forecasting, military target tracking and disaster detection etc (Goel 2001), WSNs have attracted quite attention from both academic and industrial fields in recent years. WSNs have a number of advantages over wired networks, such as ease of deployment, extended transmission range, and self-organization. There are, however, a few inherent limitations to WSNs. These include small storage capacity, limited computation resources, low communication bandwidth, and limited device energy. In terms of energy, many researchers assume that all nodes in a sensor network are battery-driven. This battery is easily irreplaceable, and become unusable after failure or energy depletion. Therefore reducing the energy consumption in sensor nodes and thereby increasing the network lifetime has become as an important and major issue in WSNs.

2.2 PROJECT DESCRIPTION

In WSNs, hierarchical network structures have the advantage of providing scalable and energy efficient solutions. For this proposed framework,

different clustering algorithms in WSNs are investigated and also compare these clustering algorithms based on the metrics such as clustering distribution, cluster's load balancing, Cluster Head's selection strategy, CH's role rotation, clusters overlapping, intra-cluster communications, reliability, security and location awareness. Furthermore, the proposed algorithm combining with periodical rotation of CHs, enable the energy load be evenly distributed among all the sensors in the network.

In addition, energy-aware algorithms for reducing the energy consumption of sensors are discussed and designed. In sensor networks, accurate data extraction is difficult. It is often too costly to obtain all sensor readings, as well as not necessary in the sense that the readings themselves only represent samples of the true state of the world. As such, one technique so-called prediction emerges to exploit the temporal correlation of sensor data. The existence of such prediction capability implies that the sensors do not need to transmit the data values if they differ from a predicted value by less than a certain prespecified threshold, or error bound. Cluster-based a localized prediction technique is introduced which is highly energy efficient and simple over the previously studied dual-prediction technique due to the reduced length of routing path for transmitting sensor data.

An optimal sleep/wake scheduling algorithm is also used, which is helpful to attain a message capture probability threshold with minimum energy consumption.

2.2.1 System Architecture and Design Issues

Depending on the application, different architectures and design goals / constraints have been considered for WSNs. Since the performance of an energy

aware protocol is closely related to the architectural model, this section strives to capture architectural issues and highlight their implications (Kemal 2011).

- **Network Dynamics:** The proposed network architectures assume that sensor nodes are stationary. On the other hand, supporting the mobility of sinks or CHs is sometimes deemed necessary. Dynamic events in the application require periodic reporting.
- **Node Deployment:** Another consideration is the topological deployment of nodes. This is application dependent. The deployment is either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data is routed through pre-determined paths. For such a static environment grid based node deployment gives better performance regarding to the coverage range and energy consumption. However in self-organizing systems, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner. For example, such a deployment can result from throwing sensor nodes from an airplane. It is assumed to be easy as well as cost-effective. When the distribution of nodes is not uniform, optimal clustering becomes a pressing issue to enable energy efficient network operation.
- **Energy Considerations:** For deterministic node deployment (square-grid), the direct routing would perform well enough since all the nodes are placed in predefined location and are reachable to the CH. Most of the time sensors are scattered randomly over an area of interest and multi-hop routing becomes unavoidable. Since the transmission power of a wireless radio is proportional to distance squared or even higher order in the presence of obstacles, multi-hop routing will consume less energy than direct communication.
- **Data Delivery Models:** Depending on the application, the data delivery model to the sink can be continuous, event-driven, and query-driven and hybrid. In the proposed model, event-driven data delivery model is used in which the transmission of data is triggered when an event occurs. The routing protocol is

highly influenced by the data delivery model, especially with regard to the minimization of energy consumption and route stability.

- **Node Capabilities:** All sensor nodes are assumed to be homogenous, having equal capacity in terms of computation, communication and power. The networks have CHs which are picked from the deployed sensors. It is assumed that the base station (BS) is more powerful than the sensor nodes in terms of energy, bandwidth and memory.
- **Data Aggregation/Fusion:** Sensor nodes consider the temporal correlation of data to send message to CH and CH performs data aggregation to send report to BS.

2.2.2 Clustering Approach

Clustering the sensor nodes as the basic of routing is an efficient mechanism for improving the energy efficiency. The application specific network topology to form cluster is proposed. In clustering schemes, redundant cluster nodes can be put into the sleep mode, since sensors within the sense and transmission range of others have no need to be active all the time. Therefore, clustering schemes are widely used in WSNs, not only due to their simple node coordination, but also because they use multi-hop routing between CHs to avoid long-range transmissions.

Cluster Head Selection

The CH selection algorithm combines energy level of the nodes and average distance of the neighbors. Initially a table is created in which it stores location information of all the nodes. When the CH selection phase starts, all the deployed nodes send their energy levels to the BS. Then on the basis of energy level, geographical area and least id CHs are selected. Network deployment is considered as manual so the base station is well informed about the geographical locations of the nodes.

Rotating the Role of Cluster Head

It is essential to rotate the role of CHs among nodes so as not to burden a few nodes with more duties than others and to easily power drained down. There are several possibilities for CH rotation. One way is to use a timer expiration to trigger the clustering algorithm. Another way is to use a dynamic parameter (e.g., remaining battery) for triggering the clustering algorithm at local regions. For example, a CH might trigger a new CH election process in its local region if its remaining battery lifetime goes below a prespecified threshold. It is obvious that more frequent CH rotation results in more clustering overhead and network interruption, while less frequent rotation may cause some nodes to die faster than others. The study of this trade-off is essential for achieving optimal network lifetime.

Algorithm 1 : Cluster Formation

Let us consider a WSN with 'n' stationary nodes randomly deployed in a [X, Y] area. The topology setup phase consists of the following steps:

```
// Initialize the network topology with specific number of nodes
1) Get Location; // Location of node in the network
2) Get Neighbors; // Check the neighbors of the node and create a table
3) Find neighbor distance; // Get the distance of neighbors and store it in the table
// CH selection
4) For each node
    If the node is alive // Energy > threshold
        Find neighboring nodes (k) having energy >= average energy threshold
        If k > 1 then
            Case 1 : //Deterministic deployment
                Determine the nodes(n) with average minimum distance from BS.
```

Case 2: //Random deployment

Determine the nodes(n) with maximum number of neighbors.

If $n > 1$ // break the tie in CH election

Determine the node(x) with least id.

- 5) Assign node x as CH
- 6) Send CH announcement to all members, BS
- // Rotate the role of CH
- 7) if timeout after m seconds
if energy level of CH < threshold then
Goto step 4;

2.2.3 Adaptive Scheme to Enable/Disable Prediction Operations

Prediction operation

In previous studies, the predictor training and prediction operations are carried out by the BS only, but not the sensor nodes, despite their increasing computing capacity. This solution while practical has many disadvantages, such as a high energy consumption incurred by transmitting the raw data to the base station, the need for wireless link bandwidth, and potential high latency. One solution which is provided in the proposed framework is clustering-based localized prediction (Xu 2003). It is expected that the use of localized prediction techniques is highly energy efficient due to the reduced length of routing path for transmitting sensor data.

Adaptive Scheme

It is noted that unlike previous dual-prediction techniques, proposed prediction operation can be enabled/disabled to achieve energy efficiency. If the prediction operation is done at all times by the CH then it could increase the communication and computation overhead and thereby the network will consume

more energy even though there is no any useful information. Therefore, in the proposed framework, it is designed to make the prediction by the CH only whenever it requires. i.e., the CH will make the prediction by using the temporal correlation among the information sent by the member nodes. As a result it says that if the correlation coefficient is too small, prediction will not be accurate. It is defined as follows (Hongbo Jiang,2011).

If the error bound ϵ satisfies $\Phi(\epsilon/m\sigma) > (k+1)2k$, the scheme with local prediction is more energy efficient. (1)

Here Φ is CDF of Gaussian white noise, m is order AR predictor, σ is the variance, k is the ratio between the communication and computation cost.

Algorithm 2 : Process at the CH

1. if timeout after $m*\Delta$ seconds
2. for each member 'i' in the cluster
3. if condition (1) holds
4. send message to member 'i' to enable prediction
5. else
6. send message to member 'i' to disable prediction
7. else
8. for each member 'i' in this cluster
9. if receive a data value from member 'i'
10. update the history data for member 'i'
11. else
12. perform prediction to update the history data

The above mentioned algorithm shows the pseudocode description of the algorithms at the cluster head. If the local prediction is enabled then the sensor nodes makes the prediction operation to identify whether the currently predicted value has to be sent to CH or not. This selective sending is based on the ϵ -loss

approximation: Given an error bound $\epsilon > 0$, a sensor node sends its value x_i to the cluster head if $|x_i - x_i'| > \epsilon$, where x_i' is a predicted representative data value to approximate the true data. The intuition of this choice is that if a past value is close to the predicted value there is not much benefit by reporting it. Because the object may not move further or it will retain in the same location. If the value is much different from the predicted value, it will be send to CH by the sensed node. For that first a localized prediction model is developed. Very complex models are not practical in the proposed framework application. Fortunately, simple linear predictors are sufficient to capture the temporal correlation of realistic sensor data as shown by previous studies (Xu 2003, McConnell 2005). The following steps shows the pseudocode description of the algorithms at each cluster member.

Algorithm 3 : Operations at the cluster members

```
if prediction is disabled or  $|x_i - x_i'| > \epsilon$ 
  send the data value to the cluster head
  update the history data using the data
else
  perform prediction to update the history data
```

2.2.4 Sleep / Awake Scheduling

A primary factor that prolongs the battery lifetime is allowing sensors to sleep when not active. This is due to the following three reasons.

- First, idle listening consumes significant energy that is comparable to transmission or reception. For example, in a Berkeley Mica2 mote, idle listening consumes energy close to that of reception. In contrast, the energy

drainage during sleep time is about three orders of magnitude less than the reception energy consumption.

- Second, battery discharge is nonlinear, and some of the unusable charges can be restored in the battery after the sleeping period.
- Third, sensors are typically deployed redundantly, which implies that not all the nodes need to be awake simultaneously.

Therefore in the proposed model, if the application requires the sensors to continuously monitor the field for unexpected events, then a CH can determine which of its cluster members are redundant and advise them to turn themselves off. Thus, a CH maintains a minimal active set of nodes in the cluster. It is also possible to elect an active set of nodes to cover the field prior to cluster formation.

The variation with sleep/wake scheduling in the proposed model is based on the following observation. For some applications, the ϵ -approximation may not be strictly required. If the confidence level (of having data values within the- error bound) is very high, e.g., above a specified threshold, say $\alpha_{\text{threshold}}$, the cluster members may never report data values to the cluster head. Therefore, there is no paramount need for the cluster members to stay awake to obtain data values most of which will be discarded anyway. To allow sleep/wake scheduling for the cluster members, the Lines (01)-(06) in Algorithm 2 is replaced by Lines (01)-(07) in Algorithm 4, and by default, disable local prediction at cluster members. When a cluster member is awake, the cluster head checks if the member's data values are within the error bound with high probability. If yes, the cluster head will send a message to power off the member. The condition should be the confidence level α_m is higher than the threshold $\alpha_{\text{threshold}}$.

However, when the cluster members sleep, the CH will not receive any data values, and hence, it is impossible to perform accurate prediction. For this

reason, periodic but infrequent collection of data from the cluster members is still necessary. In the proposed framework, only a heuristic solution to the problem is provided. Let Δ be the time interval between two consecutive reporting by a member. The duration of a sleep period is set to $p_r * \Delta$, and when a cluster member wakes up, it will continuously perform data reading (and possibly reporting) for the next $p * \Delta$ time. Initially, m_f is set to m . It can be increased if condition $\alpha_m > \alpha_{\text{threshold}}$ consistently holds, or decreased if the condition does not hold.

Algorithm 4 : A variation with sleep/wake scheduling

//sleep scheduling for members

1. **while** member 'i' is awake
2. **if** timeout after $p * \Delta$ seconds
3. **if** condition $\alpha_m > \alpha_{\text{threshold}}$ holds
4. let member 'i' power off for $p_r * \Delta$ seconds
5. **while** member 'i' is sleeping
6. **if** timeout after $p_r * \Delta$ seconds
7. awake member 'i'

When a new object arrival into the area of interest is monitored by any one of the sensor node which is in wake up state, then it immediately send a message about this arrival to its CH and then CH will further inform it to the BS. Based on the past information from the sensor node, the CH will predict the possible paths of an object moved. Using this prediction value the CH will change the state of the neighbor nodes on that possible path from sleep to wake up. Hence the sufficient number of sensor nodes available in the possible paths of an object movement only being enabled to continuously monitor the environment, thereby it reducing the power consumption of the nodes and prolongs the network lifetime.

CHAPTER 3

RESULT AND ANALYSIS

3.1 SYSTEM SPECIFICATION

3.1.1 HARDWARE REQUIREMENTS

Processor	: Pentium IV and above
Clock speed	: 550MHz
Hard Disk	: 80GB
RAM	: 128MB or above
Cache Memory	: 512KB
Monitor	: Color Monitor
Keyboard	: 104Keys

3.1.2 SOFTWARE REQUIREMENTS

Operating System	: Fedora 8
Simulator	: Network Simulator 2.34
Scripting Language	: Tcl

3.1.3 SOFTWARE DESCRIPTION

NS2 is one of the most popular open source network simulators. The original NS is a discrete event simulator targeted at networking research. First and

foremost, NS2 is an object-oriented, discrete event driven network simulator which was originally developed at University of California-Berkely. The programming it uses is C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT).

3.2 IMPLEMENTAION

To evaluate the performance value, the proposed framework was implemented using NS2 simulator.

3.2.1 Simulation Environment

There are N nodes randomly or deterministically deployed based on applications in WSN with certain range. The BS is placed inside the monitoring area. In each round, there is only one node that has a 1bit data transmission to send to the BS. All nodes will take turns to send their data with either direct transmission or multi-hop transmission using the shortest path algorithm. Network size for simulation is 1250 X 900 and number of nodes is 62. Data size is 2000 bits. Initial energy of each sensor node is set to 200J and their transmission range is 250m.

3.2.2 Snap Shots

Fig. 3.1 shows the snapshot of the test result. In which at initial phase all 61 nodes are deterministically deployed using square grid node deployment. Then, at each cluster only sufficient number of nodes that are able to monitor the new object arrival from anywhere into the network area, are set to awaken state and rest of the nodes are set to sleep state. These awoken nodes are shown in purple color and sleeping nodes are shown in white color in the simulation. In addition, CH is always awakened. Hence only 5 nodes are awaked initially and rest of the 10 nodes are sleeping in each cluster.

It is designed to generate new object from any location within/outside of the area of interest and it can be randomly moved anywhere with random speed. When new object arrival/ new event occurrence is sensed by any one of the awakened node (which is shown in red color), immediately it will be reported to the CH. Then CH further reports it to BS and makes localized prediction. As a result, CH wakes up only sufficient number of nodes available in the predicted possible paths of an object movement and it is shown in purple colored nodes nearby red colored node. Rest of the nodes awakened already is sending to sleep mode and it is shown by changing the purple color into white. This is continuously progressed for the entire network as shown in Fig. 3.2 and Fig. 3.3.

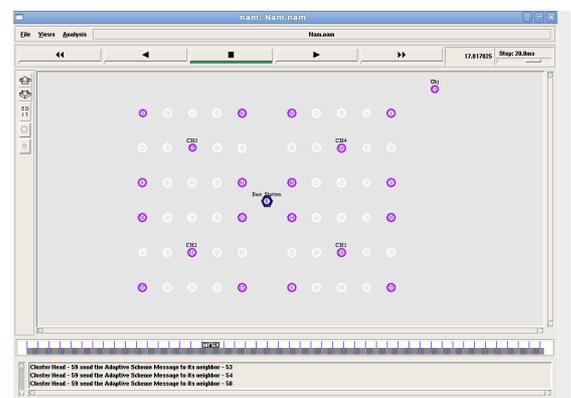


Fig. 3.1 Snapshot1 of Proposed Framework

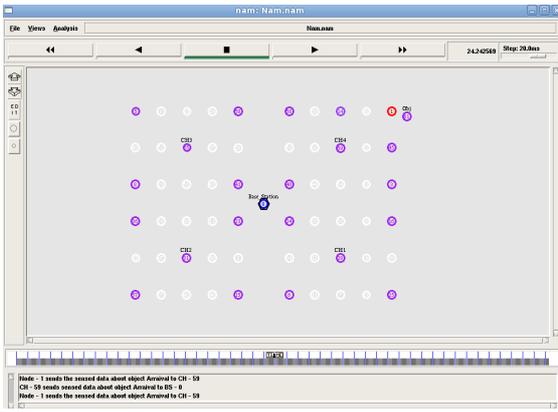


Fig. 3.2 Snapshot2 of Proposed Framework

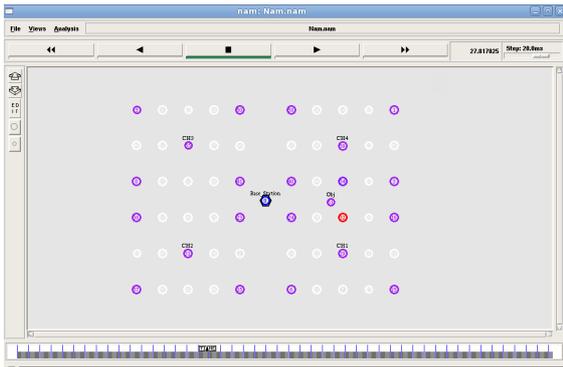


Fig. 3.3 Snapshot3 of Proposed Framework

3.3 ANALYSIS

3.3.1 Cluster Model

The CH receives data selectively reported by all of its members, and performs local prediction on the sensor data. In this way, a CH can perceive an accurate view of all sensor data across the cluster, while communication cost is drastically reduced. The graph is plotted by comparing this proposed cluster-based, adaptive prediction and sleep/awake scheduling scheme with non-cluster WSN architecture.

3.3.2 Average Energy Consumption

The energy consumption with/without adaptive scheme to control local prediction is compared. For that, only the energy consumption at all cluster members during the particular period is measured. On average, there are 4 clusters and 61 nodes. I output the sum of the energy consumption of all these 61 nodes.

3.3.3 Throughput

The performance of the algorithms is evaluated in terms of the total number of transmitted packets by all sensor nodes. For brevity, only the representative results were reported. Proposed framework will yield the scalability because the distributed techniques perform data update and prediction locally.

3.3.4 Performance Evaluation

This proposed framework is compared with non-cluster and non-adaptive sleep/awake scheduling without any localized prediction environment. A graph is plotted for this comparative study as shown in Fig. 3.4 and Fig. 3.5.

Fig. 3.4 shows the average energy consumption for all nodes at particular interval period. The remaining residual energy for all nodes in the network is considered in Y-axis and interval period of time when this measurement taken is considered in X-axis. Ex_Energy.xg specifies the non-cluster, non-adaptive scheme measurements (shown in red colored line). Energy.xg specifies the result of proposed framework (blue colored line). In a same way, Fig. 3.5 specifies the throughput of proposed framework in blue colored line (name as Throughput.xg) and throughput of non-cluster, non-adaptive scheme framework in red colored line (named as Ex_throughput.xg) by considering all packets transferred in the entire network.

Fig. 3.6 shows the average energy consumption level for the i) non-cluster, non-adaptive scheme (shown in red colored line, named as Ex.Energy.xg), ii) proposed framework (shown in green colored line, named as Energy.xg) and for iii) proposed framework while CH role is changed to other node (shown in blue colored line, named as Md_Energy.xg) in another scenario.

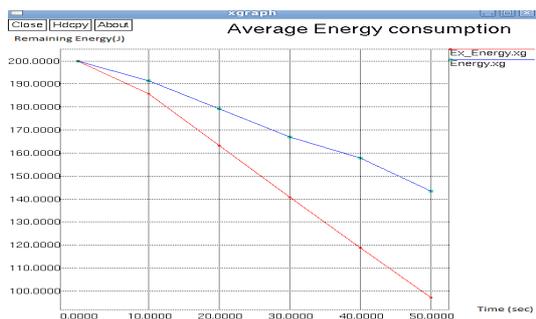


Figure 3.4 Average Energy Consumption with and without proposed schemes

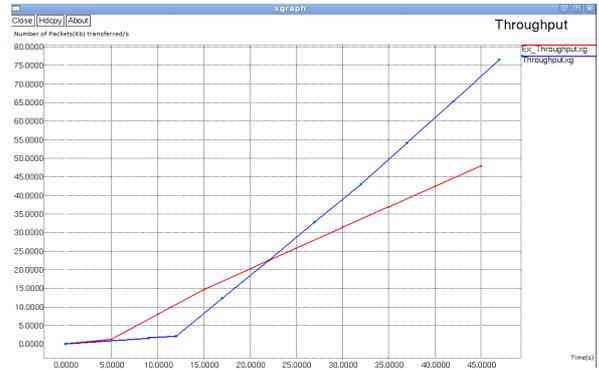


Fig. 3.5 Average Throughput with and without proposed schemes



Fig. 3.6 Average Energy Consumption in proposed scheme while CH role rotated, with and without proposed scheme

Result Analysis

From these graphs, it is clearly known that the proposed framework definitely has considerably greater remaining residual energy and provides considerably good throughput for all dynamic event generation. By considering all different scenarios in various test cases, the proposed framework provides nearly 28% (ranges from 23 to 32%) of better energy consumption on average.

3.4 CONCLUSION AND FUTURE WORK

In WSNs, since the sensor nodes are energy constrained and have limited lifetime, energy consumption of sensor nodes becomes as a major issue. Though there are several work already done to reduce the energy consumption, still it does not attain the fruitful result and research is going on. Therefore, the goal of this proposed framework is to provide a better approach to reduce the energy consumption in WSNs and to prolong the network lifetime. It is achieved by two main approaches: 1) clustering-based: sensor nodes form clusters and elect the cluster heads in such a way to improve energy efficiency, and 2) prediction based: energy-aware prediction is used to find the subtle trade-off between communication and prediction cost. The detailed analysis and description of its two main components: adaptive scheme to enable/disable prediction operations and the sleep/awake scheduling, is presented. Via performance evaluation, it is shown that it achieves approximately 28% (ranges from 23 to 32 %) energy efficiency even though the object arrived from any random location and moves randomly.

There are several future directions. The test cases may be considered with more number of new object arrivals from different directions. More precise algorithm can be developed to form cluster for dynamic changes like increasing the number of nodes in WSN, object tracking. Energy efficiency level can be computed by considering only the nodes that are having capability to transmit and receive messages. Real testing can be planned in real sensor networks. It is suggested to seek for the possibility of using more efficient algorithms to reduce the computational overhead of prediction and aggregation techniques. Other techniques such as skeleton extraction can be integrated to improve the quality of clustering.

APPENDIX

Sample Source Code

Proposed.tcl

```
#..... Environmental Settings .....
set Nn 62 ;# Number of Mobilenodes
set val(x) 1250 ;# X Co-ordinate
set val(y) 900 ;# Y Co-ordinate
set chan [new Channel/WirelessChannel] ;# Chaneel Type

#..... Simulator Object Creation .....
set nso [new Simulator]

#..... Trace File to record all the Events .....
set fpt [open Trace.tr w]
Sns trace-all $fpt
Sns use-newtrace

#..... NAM Window creation .....
set fpn [open Nam.nam w]
Sns namtrace-all-wireless $fpn $val(x) $val(y)

#..... Topology Creation .....
set tpy [new Topography]
$tpy load_flatgrid $val(x) $val(y)

#..... General Operational Director .....
create-god $Nn

#..... Node Configuration .....
Sns node-config -adhocRouting DSR \
  -lType LL \
  -macType Mac/802_11 \
  -ifqType CMUPriQueue \
  -ifqLen 500 \
  -antType Antenna/OmniAntenna \
  -propType Propagation/TwoRayGround \
  -phyType Phy/WirelessPhy \
  -topoInstance $tpy \
  -channel Chan \
```

```
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON \
-idlePower 1.2 \
-rxPower 1.0 \
-txPower 1.5 \
-sleepPower 0.000015 \
-initialEnergy 200 \
-energyModel EnergyModel

#..... Node Creation .....
for { set i 0 } { $i < $Nn } { incr i } {
  set node($i) [Sns node]
  Sns initial_node_pos $node($i) 30
  Sns color $node($i) black
}

#..... Agent Creation .....
proc Connecting-Agent { src sink pkt itv } {
  global nso
  #--- Source Agent ---
  set udp [new Agent/UDP]
  Sns attach-agent $src $udp
  #- Create a Traffic Application -
  set cbr [new Application/Traffic/CBR]
  Sns attach-agent $udp
  Sns set packetSize $pkt ;# Set Packet size
  Sns set interval $itv ;# Set Interval
  #- Attach CBR source to sink --
  Sns connect $udp $sink
  return $cbr
}

#..... Destination Agent .....
for { set i 0 } { $i < $Nn } { incr i } {
  Sns at 0.0 "$node($i) label-color black"
  set sink($i) [new Agent/LossMonitor]
  Sns attach-agent $node($i) $sink($i)
}

#..... Neighbors Distance Calculation .....
proc Distance { } {
  global node nso Nn
  set fpd [open "Distance.Cal" w]
  puts $fpd "-----"
  puts $fpd "Node\tNeighbors\ttx-cor\tty-cor\tDistance"
  puts $fpd "-----"
  for { set i 0 } { $i < $Nn } { incr i } {
```



```

if { [expr $tm%15]==0 } {
  set fp [open "ctemp" w]
  puts $fp "$cl [expr $tm-15] $tm"
  close $fp
  $nsno at $tm "exec awk -f Energy.awk ctemp Distance.Cal Trace.tr CH.Cal"
  $nsno at $tm "source Move.tcl"
  $nsno at [expr $tm+0.6] "Distance"
}
$nsno at $tm "$node(37) setdest [expr rand()*1100] [expr rand()*900] 30"
}
#---- Hop Count ----
proc hop { src snk tm int } {
  set tmp [open temp w]
  puts $tmp "$src $snk $tm $int"
  close $tmp
  exec awk -f Hop.awk temp Distance.Cal
}
$nsno at 0.05 "Deploy 0.05 1 $x $y"
$nsno at 0.50 "Deploy 0.50 0 $x $y"
$nsno at 0.6 "Distance"
for { set i 11 } { $i<51 } { incr i } { $nsno at $i "Sense $i" }
$nsno at 0.9 "$node(0) add-mark c1 black hexagon"
$nsno at 0.9 "$node(0) label Base_Station"
$nsno at 0.9 "$node(37) label Obj"
#----- Broadcast Msg -----
set tmp [open "atemp" w]
puts $tmp "1.0 0.05 0 256 0.5"
close $tmp
$nsno at 0.9 "exec awk -f Broadcast.awk atemp Distance.Call"
$nsno at 1.0 "source Broadcast.tcl"
#.....For Graph .....
set eng [open Md_Energy.xg w]
puts $eng "Markers: true"
puts $eng "BoundingBox: true"
puts $eng "0 200"
close $eng
#..... Execution NAM Window .....
proc Stop { } {
  global nso fbn node
  $nsno flush-trace
  close $fbn
  for {set i 10} {$i<60} {set i [expr $i+10]} {
    set tmp1 [open ene.tr w]
    puts $tmp1 "[expr $i-5] $i"
    close $tmp1
  }
}

```

```

exec awk -f Avg_energy.awk ene.tr Trace.tr
set eng1 [open Md_Energy.xg a]
set tmp3 [open ene.tr r]
set val1 [gets $tmp3]
puts $eng1 $val1
close $eng1 ; close $tmp3
exec xgraph Ex_Energy.xg Energy.xg Md_Energy.xg -t "Average Energy
consumption" -x "Time" -y "Remaining Energy" &
exec nam -r 5m Nam.nam &
exit 0
}
$nsno at 50.0 "Stop"
$nsno run
Broadcast.awk
BEGIN{
  p=0
  }
  {
  if(FILENAME=="atemp")
  {
    tm =$1
    itval=$2
    src =$3
    pks =$4
    itv =$5
  }
  if(FILENAME=="Distance.Call")
  {
    if($1>=0 && $1<=100)
    {
      n[p,1] =$1
      n[p,2] =$2
      n[p+,3]=$5
    }
  }
}
END {
  #----- Ascending order -----
  for(x=0;x<p;x++)
  for(y=x+1;y<p;y++)
  if(n[x,3]>n[y,3] && n[x,1]==n[y,1])
  {
    temp1=n[x,3]
    n[x,3]=n[y,3]
    n[y,3]=temp1
    temp2=n[x,2]
    n[x,2]=n[y,2]
  }
}

```

```

n[y,2]=temp2
}
x=0
y=0
a[0]=src
for(s=0;s<=x;s++) #indicate Route order
{
  src=a[s]
  for(j=0;j<p;j++)
  {
    f=1
    if(src==n[j,1])
    {
      for(s1=0;s1<=x;s1++) #check chain format
      if(a[s1]==n[j,2])
      f=0
    }
    if(f==1)
    {
      m[x,1]=n[j,1]
      m[x,2]=n[j,2]
      a[x+1]=n[j,2]
      x++
    }
  }
  for(i=0;i<x;i++)
  {
    print "set inf"i" [Connecting-Agent $node("m[i,1]") $sink("m[i,2]") "pks" "itv"] >
    "Broadcast.tcl"
    print "$nsno at "tm" \"$Sinfi" start" > "Broadcast.tcl"
    print "$nsno at "tm+itval" \"$Sinfi" stop" > "Broadcast.tcl"
    print "$nsno at "tm" \"$node("m[i,2]") color maroon" > "Broadcast.tcl"
    print "$nsno at "tm" \"$node("m[i,1]") color blue" > "Broadcast.tcl"
    print "$nsno at "tm+itval" \"$node("m[i,2]") color purple" > "Broadcast.tcl"
    print "$nsno at "tm+0.025" \"$nsno trace-annotate \\\"Node - \"m[i,1]\" send the
    Broadcast MSG to its neighbor - \"m[i,2]\" \\\" > "Broadcast.tcl"
    if(m[i,1]!=0)
    print "$nsno at "tm+itval" \"$node("m[i,1]") color purple" > "Broadcast.tcl"
    tm=tm+itval
  }
  for(x=0;x<62;x++)
  print "$nsno at "tm" \"$node("x" color purple" > "Broadcast.tcl"
  print tm+0.5 " 0.1 256 0.05 0" > "btemp"
  print "$nsno at "tm+0.5" \"exec awk -f Cluster.awk btemp Distance.Call" >
  "Broadcast.tcl"
  print "$nsno at "tm+0.5" \"source Cluster.tcl" > "Broadcast.tcl"
}

```

```

Hop.awk
BEGIN {
  i=0
  pks=256
  itv=0.05
  }
  {
  if(FILENAME=="temp")
  {
    src=$1
    des=$2
    tm=$3
    itval=$4
  }
  if(FILENAME=="Distance.Call")
  if(i==$1)
  {
    s[i,1]=$1
    s[i,2]=$3
    s[i+,3]=$4
  }
}
END {
  min=2500
  k=0
  desx=s[des,2]
  desy=s[des,3]
  a[0]=src
  for(hct=0;a[hct]!=des;hct++)
  {
    src=a[hct] # Alternative src node
    k=k+1
    srcx=s[src,2]
    srcy=s[src,3]
    for(j=0;j<j++ # find all node Distances
    if(src!=j)
    {
      x=s[j,2]
      y=s[j,3]
      srcd=int(sqrt(((x-srcx)^2)+((y-srcy)^2)))
      desd=int(sqrt(((x-desx)^2)+((y-desy)^2)))
      if(srcd<=250 && desd<min) # Check high dist from src and low dis from des
    }
  }
}

```

```

{
  min=desd
  a[k]=j
} } }
for(q=0;q<hct;q++)
{
  print "set inf"q" [Connecting-Agent Snode("a[q]") Ssink("a[q+1]") "pks" "itv"] >
  "Hop_count.tcl"
  print "Sns0 at "tm" \"Sinf"q" start\" > "Hop_count.tcl"
  print "Sns0 at "tm+itval" \"Sinf"q" stop\" > "Hop_count.tcl"
  tm=tm+0.01
}
if(a[0]!="")
  print "" > "Hop_count.tcl"
}
Cluster.awk
BEGIN {
  i=0
  {
    {
      if(FILENAME=="btemp")
      {
        tm =$1
        itval=$2
        pks =$3
        itv =$4
        flg =$5
      }
      if(FILENAME=="Distance.Cal" )
      if($1>=0 && $1<=100)
      {
        n[i,1] =$1
        n[i,2] =$2
        n[i++,3]=$5
      }
    }
  }
  END {
    #----- Get Neighbor Details -----
    nd=0
    cn=0
    j=0
    for(x=0;x<i;x++)
    {
      if(nd!=n[x,1])
      {

```

```

      nnd[j++]=cn
      nd=n[x,1]
      cn=0
    }
  }
  cn++
}
nnd[j]=cn
#----- Get CH Using Max Neighbor -----
k=0
for(x=1;x<=j;x++)
if(nnd[x]>=8)
ch[k++]=x
for(x=0;x<k;x++)
{
  print "Sns0 at "tm" \"Snode("ch[x]") label CH"x+1\" > "Cluster.tcl"
  print ch[x] > "CH.Cal"
  for(y=0;y<i;y++)
  if(n[y,1]==ch[x])
  {
    print "set inf"y" [Connecting-Agent Snode("ch[x]") Ssink("n[y,2]") "pks" "itv"] >
    "Cluster.tcl"
    print "Sns0 at "tm" \"Sinf"y" start\" > "Cluster.tcl"
    print "Sns0 at "tm+itval" \"Sinf"y" stop\" > "Cluster.tcl"
    if(flgs=0)
      print "Sns0 at "tm+0.025" \"Sns0 trace-annotate \\\"Cluster Head - "ch[x]" send
      the CHADV Message to its neighbor - "n[y,2]" \\\" > "Cluster.tcl"
    if(flgs=1)
      print "Sns0 at "tm+0.025" \"Sns0 trace-annotate \\\"Cluster Head - "ch[x]" send
      the Adaptive Scheme Message to its neighbor - "n[y,2]" \\\" > "Cluster.tcl"
    tm=tm+itval+0.01
  }
}
#----- Next Process -----
if(flgs=0)
{
  print tm+0.5" 0.1 256 0.05 1" > "btemp"
  print "Sns0 at "tm+0.5" \"exec awk -f Cluster.awk btemp Distance.Cal\" >
  "Cluster.tcl"
  print "Sns0 at "tm+0.5" \"source Cluster.tcl\" > "Cluster.tcl"
}
if(flgs=1)
{
  for(x=0;x<i;x++)
  if(n[x,1]!=0 && n[x,3]<202)
  for(y=0;y<k;y++)
  if(n[x,1]==ch[y])

```

```

  print "Sns0 at "tm" \"Snode("n[x,2]") color white\" > "Cluster.tcl"
} }
Avg_Energy.awk
BEGIN {
  k =0
  exh=0
  rem=0
}
{
  if(FILENAME=="ene.tr")
  {
    st=$1
    et=$2
  }
  if(FILENAME=="Trace.tr")
  if($3 >=st && $3 <= et && $1=="N")
  {
    exh=exh+$7
    k++
  }
}
END {
  if(k==1)
  avg=0
  else
  {
    avg=exh/(k-1)
    print et" "avg > "ene.tr"
  }
}
Energy.awk
BEGIN {
  a=0
  i=0
  j=0
  m=0
}
{
  if(FILENAME=="ctemp")
  {
    ch=$1
    st=$2
    et=$3
  }
  if(FILENAME=="Distance.Cal")
  {
    if($1>=0 && $1<100)

```

```

  {
    ndn[i,1] =$1
    ndn[i,2] =$2
    ndn[i++,3]=$5
  }
  if($1==j)
  {
    node[j,1] =$1
    node[j,2] =$3
    node[j++,3]=$4
  }
  if(FILENAME=="Trace.tr")
  if($3 >=st && $3 <= et && $1=="N")
  {
    ne[$5]=$7
    if($5>max)
    max=$5
  }
  if(FILENAME=="CH.Cal")
  cht[a++]=$1
}
END {
  #----- Find Cluster -----
  for(k=0;k<i;k++)
  if(ndn[k,1]==ch)
  cl[m++]=ndn[k,2]
  cl[m]=ch
  #--- Find New Cluster Head ---
  emax=0
  for(k=0;k<m;k++)
  {
    print "Sns0 at "et" \"Snode("cl[k]") color purple\" > "Move.tcl"
    for(x=0;x<=max;x++)
    if(cl[k]=x && emax<ne[x])
    {
      emax=ne[x]
      nch=x
    }
  }
  for(k=0;k<a;k++)
  if(cht[k]==ch)
  print nch > "CH.Cal"
  else
  print cht[k] > "CH.Cal"
  print "Sns0 at "et" \"Snode("nch") label NCH\" > "Move.tcl"
  print "Sns0 at "et" \"Snode("ch") label .\" > "Move.tcl"

```

```

for(k=0;k<j;k++)
{
if(k==ch)
{
print "Sns0 at "et+0.1" \"Snode("nch") setdest "node[k,2]" "node[k,3]" 3000\" >
"Move.tcl"
print "Sns0 at "et+0.4" \"Snode("nch") setdest "node[k,2]" "node[k,3]" 3000\" >
"Move.tcl"
}
if(k==nch)
{
print "Sns0 at "et+0.1" \"Snode("ch") setdest "node[k,2]" "node[k,3]" 3000\" >
"Move.tcl"
print "Sns0 at "et+0.4" \"Snode("ch") setdest "node[k,2]" "node[k,3]" 3000\" >
"Move.tcl"
}
}
}
}
}

```

REFERENCES

1. Baranidharan,B., Shanthi,B., "A Survey on Energy Efficient Protocols for Wireless Sensor Networks", IJCA, vol.11, 2010.
2. Bo-Chao Cheng, Hsi-Hsun Yeh, and Ping-Hai Hsu,"Schedulability Analysis for Hard Network Lifetime Wireless Sensor Networks With High Energy First Clustering", IEEE Transactions on reliability, 2011.
3. Chu,D., Deshpande,A., Hellerstein,J.M., and Hong,W., "Approximate Data Collection in WSNs Using Probabilistic Models," Proc.IEEE(ICDE), 2006.
4. Gedik,B., Liu,L., and Yu,P.S., "ASAP: An Adaptive Sampling Approach to Data Collection in WSN," IEEE Trans. Parallel and Distributed Computing, vol. 18, no. 12, pp.1766-1783, 2007.
5. GoeI,S., and Imielinski,T., "Prediction-Based Monitoring in Sensor Networks: Taking Lessons from MPEG," ACM SIGCOMM Computer Comm. Rev., vol. 31, no. 5, pp. 82-98, 2001.
6. Hongbo Jiang, Shudong Jin, Chonggang Wang, Senior Members, IEEE,"Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks", IEEE Transaction on Parallel and Distributed Systems, vol.22, No.6, 2011.
7. Heinzelman,W.P., Chandrakasan,A.P., and Bala krishnan,H., "An Application-Specific Protocol Architecture for WSN," IEEE Trans. Wireless Comm., vol. 1, no. 4, pp. 660-670, 2002.
8. Meka,A., and Singh,A.K., "Distributed Spatial Clustering in Sensor Networks," Proc. Int'l Conf. Extending Database Technology (EDBT), 2006.
9. McConnell,S.M., and Skillicorn,D.B., "A Distributed Approach for Prediction in Sensor Networks," Proc. SIAM Int'l Conf. Data Mining Workshop WSN, 2005.
10. Sanjay Waware, Dr. Nisha Sarwade , Pallavi Gangurde,"A Review of Power Efficient Hierarchical Routing Protocols in WSNs", IJERA,vol.2, 2012.
11. Wint Yi Poe, Jens,B., Schmitt, "Node Deployment in Large Wireless Sensor Networks: Coverage, Energy Consumption,and Worst-Case Delay", AINTEC'09, November 18-20,Bangkok, Thailand, 2009.
12. Wu,Y., Fahmy,S., and Shroff,N.B., "Energy Efficient Sleep/Wake Scheduling for Time-Synchronized Multi-hop WSNs", Technical Report, 2006.

13. Wu,Y., Fahmy,S., Shroff,N.B., "Optimal sleep/wake scheduling for time-synchronized WSNs with qos guarantees", in Proc. Of IEEE IWQoS,2006.
14. Xu,Y., and Lee,W.C., "On Localized Prediction for Power Efficient Object Tracking in WSNs,"Proc. IEEE,(ICDCS), 2003.

LIST OF PUBLICATIONS

1. M.Vijayalakshmi, Dr.V.Vanitha, "Energy Efficient Cluster Based Data Collection for Wireless Sensor Networks" , Fourth National Conference on Innovations in Information Technology (NCIIT 2013), Bannari Amman Institute of Technology, Sathyamangalam, 21.02.2013.
2. M.Vijayalakshmi, Dr.V.Vanitha, "Cluster based Adaptive Prediction Scheme for Energy Efficiency in Wireless Sensor Networks" was selected to publish in the Conference on Computational Intelligence and Information Technology (CCIIT 2013) going to be held at Info Institute of Technology on 6.5.2013.