ENERGY EFFICIENT SERVICE-ORIENTED ROUTING OF QUERIES IN WIRELESS SENSOR NETWORKS

PROJECT REPORT

Submitted by

SURIYA PRABA T

Reg. No: 0920108023

In partial fulfillment for the award of the degree

of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 049

APRIL 2011

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 049

Department of Computer Science and Engineering

PROJECT WORK

APRIL 2011

This is to certify that the project entitled

ENERGY EFFICIENT SERVICE-ORIENTED ROUTING OF QUERIES IN WIRELESS SENSOR NETWORKS

This the bonafide record of project work done by

SURIYA PRABA T

Register No: 0920108023

of M.E. (Computer Science and Engineering) during the year 2010-2011.

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 25/04/2011

Internal Examiner

External Examiner

DECLARATION

I affirm that the project work titled **ENERGY EFFICIENT SERVICE-ORIENTED ROUTING OF QUERIES IN WIRELESS SENSOR NETWORKS** being submitted in partial fulfillment for the award of M.E Computer Science and Engineering degree is the original work carried out by me. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

SURIYA PRABA T

Register No: 0920108023

I certify that the declaration made above by the candidate is true

Mrs. **V.VANITHA M.E.,**

Associate Professor

Department Of Computer Science and Engineering,
Kumaraguru College of Technology,
(An Autonomous Institution)
Coimbatore-641 049.

SRI SHAKTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY

POWERING THE YOUTH EMPOWERING THE NATION

TÜVRheinland CERT ISO 9001

Approved by AICTE, New Delhi ■ Affiliated to Anna University of Technology

'Sri Shakthi Nagar', L&T By-Pass, Coimbatore - 641 062. Ph. : 0422 - 6450891 / 92 / 93 E-mail : youcanreachus@siet.ac.in Web : www.siet.ac.in

International Conference on Computer Communication and Informatics

It is certified that

T. Suriya Praba

has presented a technical paper on

Network structure based routing Techniques in wireless Sensor Networks.

during the "International Conference on Computer Communication and Informatics" conducted during January 11th and 12th, 2011.

Organizing Secretary | Session Chair | Principal | Chairman

MAHENDRA
EDUCATIONAL INSTITUTIONS

MAHA COLLEGE OF ENGINEERING

Approved by AICTE, Affiliated to Anna University of Technology, Coimbatore.
Salem - Chennai Highway, Minnampalli, Salem - 636 106.

NCETECE'11

Certificate

This is certified that Mr./Ms./Mrs. T.SURIYA PRABA

has participated and presented a paper entitled ENERGY EFFIENCIT SERVICE ORIENTED ROUTING OF QUERIES WSN in the National Conference on Emerging trends in Electronics & Communication Engineering on 1-4-2011 in Maha College of Engineering, Salem.

Convener

Principal

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 our Chairman **Padmabhusan Arutselvar Dr.N.Mahalingam B.Sc., F.I.E.,** for giving this opportunity to pursue this course.

I would like to thank **Dr.S.Ramachandran Ph.D.,** *Principal* for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Mrs.P.Devaki M.E.,** *Head of the Department,* Computer Science and Engineering, for his precious suggestions.

I thank all project committee members for their comments and advice during the reviews.

I register my hearty appreciation to the Guide **Mrs.V.Vanitha M.E.,** *Associate Professor,* Department of computer Science and engineering, my thesis advisor. I thank for her support, encouragement and ideas. I thank her for the countless hours she has spent with me, discussing everything from research to academic choices.

I would like to convey my honest thanks to all **Teaching** staff members and **Non Teaching** staffs of the department for their support. I would like to thank all my classmates who gave me a proper light moments and study breaks apart from extending some technical support whenever I needed them most.

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

TABLE OF CONTENTS

CONTENTS **PAGE NO**

ABSTRACT

Wireless Sensor Network (WSN) is a network of few hundred to several thousands of autonomous nodes, which work cooperatively to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. The main issue of WSN is energy constraint as the node drains faster and the battery is also not rechargeable and replaceable. Analysis on wireless sensor network shows that communication module is the main part which consumes major portion of the sensor nodes energy. Since routing protocols directly access the communication module the design of energy efficient routing protocols is a challenging task. Previously proposed query routing schemes are either not Service based or lack to take advantage of a hierarchical organization of the network when they are service based. In this paper Energy efficient **S**ervice oriented **Q**uery **R**outing (ESQR) scheme is proposed to achieve data centric routing of queries in Wireless Sensor Networks. The routing scheme works by forming service clusters of the nodes based on three different parameters namely nodes distance to base station, residual energy and service provided by the nodes. The queries are routed through the cluster heads via regional cluster heads to reach the nodes that can provide the requested service. Simulation result shows that ESQR performs better than the other algorithms in most of the cases. Therefore, ESQR is a stable and energy efficient routing algorithm to be utilized in any real time WSN application. This approach saves energy, extends network lifetime, without duplicating a routing table on each node and improves the latency in routing queries.

ஆய்வுச்சுருக்கம்

கம்பியிலா உணரி வலையமைப்பானது ஆயிரக்கணக்கான முனைகளைக் கொண்டு சுற்றுச்சூழள் சூழ்நிலைகளை கண்காணிப்பதற்காகப் பயன்படுத்தப்படுகிறது.இதன் முக்கிய குறைபாடு இதன் மின் கலத்தை நாம் மின்னூட்டு செய்யவோ அல்லது மற்றவோ முடியாது.எனவே இதன் ஆற்றலை உரிய முறையில் பயன்படுத்த வேண்டும்.

இதில் தொடர்புப்பகுதியே அதிக அற்றலை பயன்படுத்துவதாக ஆய்வுகள் தெரிவிக்கின்றன.எனவே இந்த ஆய்வில் அற்றலை திறமைமிக்க வழியில் பயன்படுத்தும் வினா வழியமைக்கும் வரைமுறை அறிமுகப்படுத்தப்படுகிறது.

இந்த இலக்கை எட்டுவதற்க்காக ஒரே சேவையை கண்காணிக்கும் முனைகள் ஒரே கூட்டமாகக் கருதப்படுகிறது.மேலும் இதற்க்கென்று ஒரு கூட்டத் தலைமுனையும் நியமிக்கப்படுகிறது.மேலும் இந்த கூட்டமைப்பானது மீதமுள்ள அற்றல் மற்றும் அடிப்படை நிலையத்திலிருந்து அதன் தொலைவு ஆகியவற்றையும் கருத்தில் கொண்டு உருவாக்கப்படுகிறது.

அடிப்படை நிலையத்தில் வினாக்கள் சமர்பிக்கப்படும்பொழுது அவைகள் கூட்டத்தலை முனை வழியாக சரியான முனைக்கு வழிப்படுத்தப்பட்டு தேவையான சேவையானது பெறப்படுகிறது.இந்த அமைப்பின் மூலம் முனைகளின் அற்றல் சேமிக்கப்படுவதுடன் அவைகளின் ஆயுட்காலமும் நீட்டிக்கப்படுகிறது.மேலும் இதில் வழியமைப்பு நகல் அட்டவணையை பேணுவதற்கான தேவை ஏற்படுவதில்லை.

LIST IF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Networks
ADC	Analog to Digital Converter
QoS	Quality of Services
LEACH	Low-Energy Adaptive Clustering Hierarchy
MAC	Media Access Control
BS	Base Station
CH	Cluster Head
PEGASIS	Power Efficient GAthering in Sensor Information System
HEED	Hybrid Energy Efficient Distributed Clustering
SARP	Service Aware Routing Protocol
SQR	Service Driven Query Routing
ESQR	Energy Efficient Service-Oriented Routing of Queies

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF WIRELESS SENSOR NETWORKS

1.1.1 Introduction to Wireless Sensor Networks

Sensor networks are highly distributed networks of small, lightweight wireless node, deployed in large numbers to monitor the environment or system by the measurements of physical parameters such temperature, pressure or relative humidity. Each node of the sensor network consist of three subsystems: the sensor subsystem which senses the environment, the processing subsystem which performs local computation on the sensed data and the communication subsystem which is responsible for message exchange with neighboring sensor nodes.

The sensor nodes in a wireless sensor network are usually deployed randomly inside the region of interest or close to it. A remote *base station* (BS) connected to the Internet is engaged to give commands to all the sensor nodes and gather information from the sensor nodes. In addition to sensing, the wireless sensor nodes can process the acquired information, transmit messages to the BS, and communicate to each others. A simple architecture of the wireless sensor network is depicted in Figure. 1.1

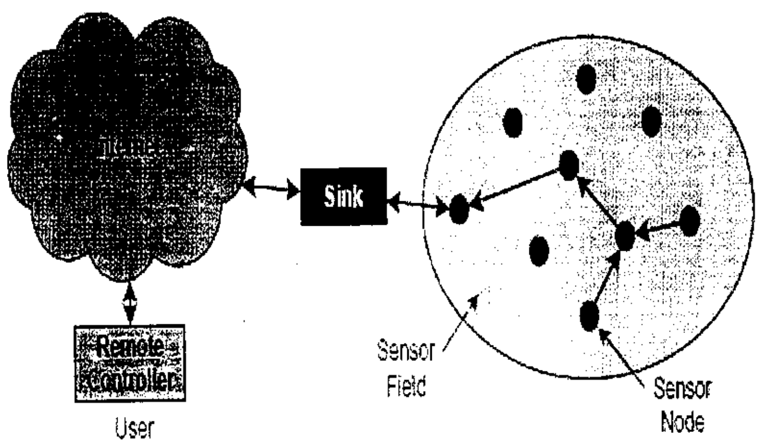


Figure 1.1The architecture of a wireless sensor network in which the sensor nodes are deployed randomly into the interested area (sensor field) and the BS (sink) connects to the Internet.

1.1.2 Components of Sensor Networks

Every sensor node is equipped with the following components:

- **Sensing unit**

  Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs).The analog signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit.

- **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 Unit**

  A transceiver unit connects the nodes to the networks. It is capable of transmitting and receiving data.

- **Power Unit**

  Every sensor node is equipped with a battery that supplies power to remain in active

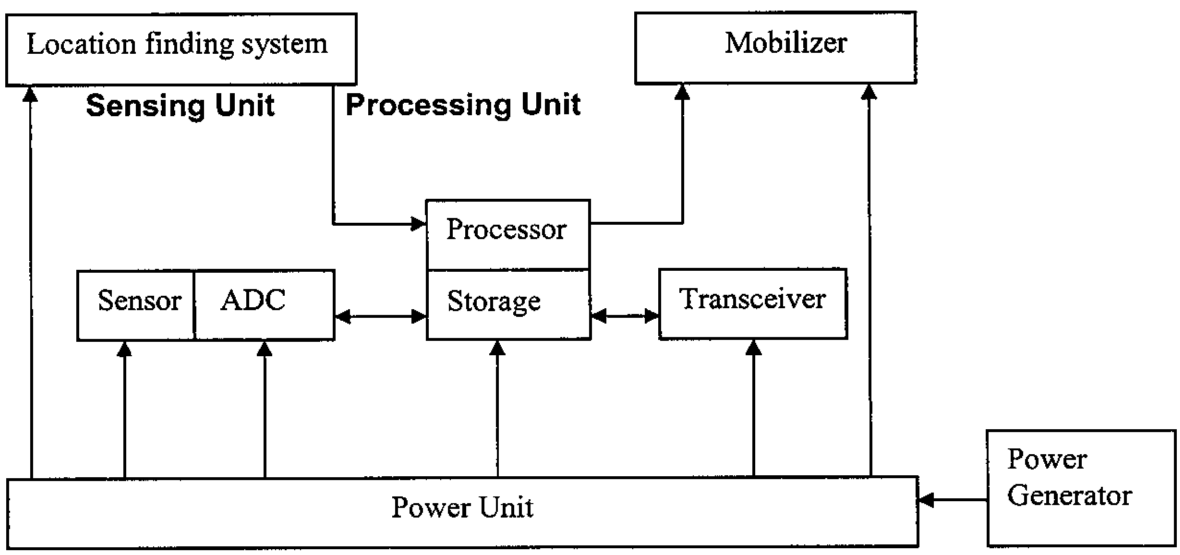


Figure 1.2 Components of a sensor node

1.1.3 Wireless Sensor Networks Architecture

The architecture of wireless sensor networks is classified into two types. They are

- Layered Architecture
- Clustered Architecture

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 figure 1.3

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
- MAC Protocol
- Routing Protocol

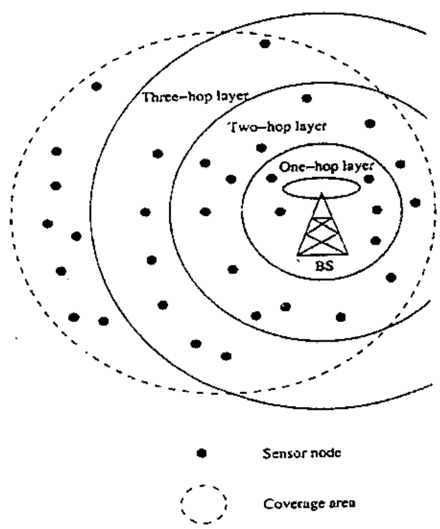


Figure 1.3 Layered Architecture in wireless sensor networks

Network Initialization & Maintenance Protocol:

- BS broadcasts its ID using CDMA common control channel (BS reaches all nodes in one hop)
- Nodes record BS ID & send beacon signal with their own IDs at their low default power levels
- All nodes the BS can hear are at 1-hop distance
- 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-layer 2
- The layer one nodes inform the BS of the layer 2
- The BS broadcasts the layer2 nodes IDs
- **To maintain**: periodic beaconing updates are required.

MAC protocol

- A Time Division CDMA (TCDMA) protocol for spatial bandwidth reuse
- Ensures a scheduling scheme for fair access.

Routing Protocol:

- Downlink from the BS is by direct broadcast on the control channel
- Enables multi-hop data forwarding to the BS
- The remaining energy is considered when forwarding to the next hop (layer)
- Only the nodes of the next layer need to be maintained in the routing table

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 figure 1.4.

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.

- ➢ Setup phase: organizing the clusters

➢ Steady-state phase: deals with the actual data transfers to the BS

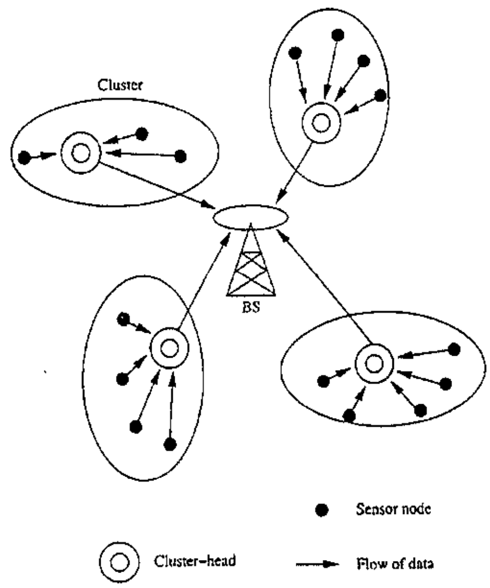


Figure 1.4 Clustered Architecture in wireless sensor networks

Setup phase:

- 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

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

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:

- All source nodes send their data to their cluster heads
- Cluster heads perform data aggregation/fusion through local transmission
- Cluster heads send them back to the BS using a single direct transmission

- After a certain period of time, cluster heads are selected again through the set-up phase
- Merits:
  - Accounting for adaptive clusters and rotating cluster heads
  - Opportunity to implement any aggregation function at the cluster heads
- Demerits:
  - Highly dynamic environments
  - Continuous updates
  - Mobility

1.1.4 Applications of Sensor Networks

Wireless Sensor Networks have a wide range of applications such as,

1. Military applications
   i. Monitoring friendly forces and equipment.
   ii. Battlefield surveillance.
   iii. Nuclear, biological and chemical attack detection.
2. Environmental Applications
   i. Forest fire detection.
   ii. Bio-complexity mapping of the environment.
   iii. Flood detection.
3. Health applications
   i. Tele-monitoring of human physiological data.
   ii. Tracking and monitoring doctors and patients inside a hospital.
   iii. Drug administration in hospitals.
4. Home application
   i. Home automation.
   ii. Smart environment.

In order to enable reliable and efficient observation and initiate right actions, physical phenomenon features should be reliably detected/estimated from the collective information provided by sensor nodes. Moreover, instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. Hence, 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 protocols in terms of energy consumption.

1.1.5 Issues and Challenges of Wireless Sensor Networks

A sensor network design is influenced by many factors, which include,

- **Energy efficiency/system lifetime**

As sensor nodes are battery-operated, protocols must be energy-efficient to maximize system life time. System life time can be measured such as the time until half of the nodes die or by application-directed metrics, such as when the network stops providing the application with the desired information about the phenomena.

- **Fault Tolerance**

Some sensor nodes may fail or be blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures.

- **Scalability**

The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. Depending on the application, the number may reach an extreme value of millions. The new schemes must be able to work with this number of nodes.

- **Production Costs**

Since the sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the networks. If the cost of the network is more expensive than deploying traditional sensors, then the sensor network is not cost-justified. As a result, the cost of each sensor node has to be kept low.

- **Environment**

Sensor nodes are densely deployed either very close or directly inside the phenomenon to be observed. Therefore, they usually work unattended in remote geographic areas. They may be working in busy intersections, in the interior of large machinery, at the bottom of an ocean, inside a twister, on the surface of an ocean. They work under high pressure in the bottom of an ocean, in harsh environments such as debris or a battlefield, under extreme heat and cold such as in the nozzle of an aircraft engine or in arctic regions, and in an extremely noisy environment such as under intentional jamming.

- **Hardware Constraints**

A sensor node is made up of four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit. Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs). The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit, which is generally associated with a small storage unit, manages the procedures that enable the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units may be supported by a power scavenging unit such as solar cells. There are also other subunits, which are application dependent. Most of the sensor network routing techniques and sensing tasks require the knowledge of location with high accuracy.

- **Sensor Network Topology**

Sheer numbers of inaccessible and unattended sensor nodes, which are prone to frequent failures, make topology maintenance a challenging task. Hundreds to several thousands of nodes are deployed throughout the sensor field.

a) Pre-Deployment Phase

Sensor nodes can be either thrown in mass or placed one by one in the sensor field. They can be deployed by dropping from a plane, delivering in an artillery shell, rocket or missile, throwing by a catapult , placing in factory, and placing one by one either by a human or a robot.

Although the sheer number of sensors and their unattended deployment usually preclude placing them according to a carefully engineered deployment plan, the schemes for initial deployment must reduce the installation cost, eliminate the need for any pre-organization and preplanning, increase the flexibility of arrangement, and promote self-organization and fault tolerance.

b) Post-Deployment Phase

After deployment, topology changes are due to change in sensor nodes position, reach ability (due to jamming, noise, moving obstacles, etc.), available energy, malfunctioning, and task details. Sensor nodes may be statically deployed. However, device failure is a regular or common event due to energy depletion or destruction. It is also possible to have sensor networks with highly mobile nodes. Besides, sensor nodes and the network experience varying task dynamics, and they may be a target for deliberate jamming. Therefore, sensor network topologies are prone to frequent changes after deployment.

c) Re-Deployment of Additional Nodes Phase

Additional sensor nodes can be re-deployed at any time to replace the malfunctioning nodes or due to changes in task dynamics. Addition of new nodes poses a need to re-organize the network. Coping with frequent topology changes in an ad hoc network that has myriads of nodes and very stringent power consumption constraints requires special routing protocols.

- **Transmission Media**

In a multi-hop sensor network, communicating nodes are linked by a wireless medium. These links can be formed by radio, infrared or optical media. To enable global operation of these networks, the chosen transmission medium must be available worldwide. One option for radio links is the use of *Industrial, Scientific and Medical* (ISM) bands, which offer license free communication in most countries.

- **Power Consumption**

The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source. In some application scenarios, replenishment of power resources might be impossible. Sensor node lifetime, therefore, shows a strong dependence on battery lifetime. The malfunctioning of few nodes can cause significant topological changes and might require rerouting of packets and re-organization of the network. Hence, power conservation and power management take on additional importance. It is for these reasons that researchers are currently focusing on the design of power aware protocols and algorithms for sensor networks. In sensor networks, power efficiency is an important performance metric, directly influencing the network lifetime. Application specific protocols can be designed by appropriately trading off other performance metrics such as delay and throughput with power efficiency. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains: *sensing*, *communication*, and *data processing*.

1.2 SERVICE-ORIENTED SENSOR NETWORKS

1.2.1 Limitations of current Sensor network architectures

Current architectures for sensor networks have inherent limitations that may be summarized as follows:

a) Tight coupling between sensor networks and applications

A tight network application coupling characterizes most current sensor network and sensor networks architectures. This coupling usually takes one of two forms:

(i) Network-dependent application development

Most current sensor applications are designed and implemented to be used on a specific sensor network or a specific type of sensor networks with specific characteristics and a specific querying interface. Often, application developers need to know network-specific information such as network topology, nodes' transmission range and processing/memory capabilities, etc.

(ii) Application-dependent network design and deployment

In some cases, a decision is first made to use a specific software system to build a sensor application. The requirements of the software system must then be taken into consideration when designing and deploying the sensor network supporting that system.

b) Costly optimization or suboptimal efficiency

In current sensor networks, application-dependent optimization makes the sensor network unable to provide the same levels of performance to other applications. Also, several rounds of optimization may be needed as the application evolves or is replaced. The alternative of application independent optimization is often too generic and does not exploit optimization opportunities that a specific class of applications may offer. Typically, this leads to suboptimal efficiency.

c) Limited reusability

Ideally, for a sensor networks to be cost-effective, it would be necessary to amortize its deployment and maintenance cost by sharing its functionalities amongst a large group of users and applications. This reusability is generally not easily achievable in current sensor infrastructures due to the tight coupling between networks and applications.

d) Low return on investment

This drawback follows from the previous one. The monolithic, application-specific Design of current sensor networks makes it difficult to reuse most of an application's modules in developing another application. Often, intensive programming is required each time a new application has to be developed.

e) Non-scalability

Most of today's sensor networks applications are designed and optimized for light loads, i.e., destined to be used by a small group of users. For example, only a few officers at the DPHE may be entitled to access the pollution monitoring system. As a result, current sensor networks often do not scale to support large numbers of simultaneous users.

1.2.2 Design objectives for Service-Oriented Sensor Networks

To enable the previous scenarios, future sensor network infrastructures must provide the following features:

a) Abstraction

In Service-Oriented Sensor Networks, programmers/users will only have to understand the interaction with the service they are requesting from the sensor–actuator network. Users would be able to specify services without necessarily knowing which sensor network(s) will actually carry out the sensing and actuation tasks that are necessary for the provisioning of those services.

b) Interoperability, composability, and retaskability

Interoperability refers to the ability of services, of possibly different providers, to readily interact with each other. This is possible because services are self-contained and expose standard interfaces to their clients. Interoperability enables services to be dynamically composed. This, in turn, enables retaskability, i.e., the ability to specify new tasks and compose new applications achieving those tasks using services that are already deployed. Retaskability eliminates the need for frequent reprogramming of sensor networks.

c) Scalability

As Service-Oriented Sensor Networks are expected to be accessible to a large number of users, the architecture must be scalable. In particular, it must provide efficient query optimization mechanisms that eliminate the traditional one-to-one mapping between users' invocations of services and the actual queries submitted to the sensor networks.

d) Decoupling between services and sensor networks

A service may be provided through different types of sensor–actuator nodes. The decoupling between services and physical nodes allows greater flexibility in maintaining sensor networks. For example, the provider of a sensor network may decide to Change the number, type, or location of some or all of the sensors used in the network without modifying the services' interfaces.

e) Better resource utilization

Consider a service for querying a sensor network. The service may receive a large number of similar or related requests in a relatively short period of time. Using mechanisms such as request similarity detection and caching, the service may be able to answer most of the requests without actually querying the sensor network. This will reduce energy consumption for the sensor network and improve response time for the service's clients.

f) Resilience

As a service may be provided through several sensor networks, applications will become less dependent on the state of specific sensor networks.

g) Cost reduction

Sensor–actuator services are expected to be reusable and often generic . This has the advantage of lowering the cost of developing sensor-actuator applications. Moreover, as sensor services proliferate, service providers will have to compete for clients. Consequently, sensing and actuation services will increasingly be provided to users at a lower cost and/or higher quality

1.2.3 Service-Oriented Sensor network

In Service-Oriented Sensor networks nodes' sensing and actuation capabilities are exposed to applications in the form of a collection of in-network services. Services in the proposed approach are lightweight code units deployed directly on top of the operating system of nodes. Each service provides some sensing or actuation functionality supported by the local node. Services may be individually invoked or combined in complex ways to form a ''virtual'' sensor networks with far richer sensing and actuation capabilities. Applications invoke services using a service oriented query model that offers a high abstraction level and, yet, enables a wide spectrum of low-level optimization mechanisms. A prototype for service-oriented query processor was implemented on top of Tiny OS. Initial results show significant advantages of the proposed approach as a flexible programming model and an efficient query optimization technique for sensor networks.

1.2.4 An Architecture for Service Oriented Sensor Networks:

This section describes a service-oriented architecture for nodes that supports the ECAST query model. Figure 7 shows an overview of this architecture. The software running on top of the operating system at each node is organized into two layers:

(i) **Service Layer:**

A node in a Service-Oriented Sensor Network runs a collection of lightweight services. Each service is a software module that carries out some sensing or control function. Conceptually, a service is a function that receives input values called parameters and that returns output values called results. Without loss of generality, the discussion was illustrated through sensing services that accept no parameters. The syntax getAttribute() is adopted here to express invocations of sensing services. For example, an invocation of a sensing service that samples and returns temperature would be Noted as getTemperature().

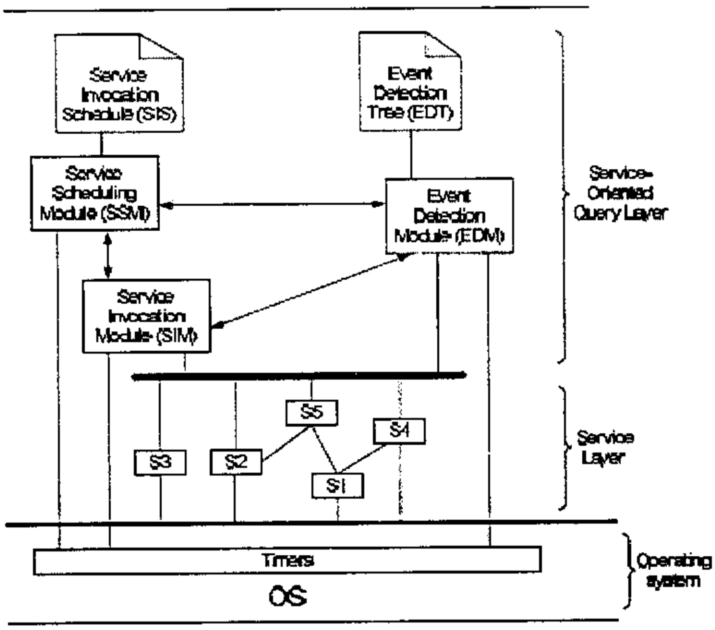


Figure 1.5: Nodes architecture in Service Oriented Sensor Networks

A service running on a node may be invoked by the base station or by other nodes of the Sensor Network. Service invocations are asynchronous, i. e., the issuer of the service invocation is not blocked when it issues a service invocation. A service may invoke another service on the same node or interact directly with OS components (e. g., sensor controllers, and timers) of its local node. These components, in turn, interact with the node's hardware modules (e. g., actuation unit, clock, etc.)

(ii) Service-Oriented Query Layer:

This layer consists of three modules:

• Event Detection Module (EDM): This module detects the events that are relevant to the current query load at the local node.

• Service Scheduling Module (SSM): This module schedules repetitive queries according to their frequencies and expiration time. It maintains a list of services to be invoked and the times of invocation in a service invocation schedule (SIS).

• Service Invocation Module (SIM): This module receives requests for service invocation from the EDM and the SSM. It then invokes the service and, if applicable, returns the results of the invocation to the query issuer.

1.3 ROUTING IN SENSOR NETWORKS:

Routing in wireless sensor network is the process of forwarding the queries from base station to the node which is providing required service. Wireless sensor networks are large scale networks consisting of a large number of tiny sensor nodes and a few base stations, which communicate using multi-hop wireless communications. Analysis on WSNs shows that communication module is the main part which consumes most of the sensor energy. Since routing protocols and MAC protocols directly access the communication module the design of energy efficient routing protocols for such networks is a challenging task, which has been in the focus of the sensor network research community in the recent past. This effort resulted in a huge number of sensor network routing protocols. The efficiency of sensor networks strongly depends on the routing protocol used.

Each node in a sensor network is composed of a radio-transducer, a small microcontroller and a long lasting battery for energy source. Primarily these sensors are used for data acquisition and are required to disseminate the acquired parameters to special nodes called sinks or base-stations over the wireless link as shown in figure 1.1. The base-station or sink collects data from all the nodes, and then analyzes this data to draw conclusions about the on-going activity in the area of interest [1]. Sinks or base- stations being powerful data processors can act as gateways to other existing communications infrastructure or to the Internet where a user can have access to the reported data.

All nodes in a network communicate with each other via wireless links, where the communication cost is much higher than the computational cost. Moreover, the energy needed to transmit a message is about twice as great as the energy needed to receive the same message. Unfortunately, some requirements for the routing protocols are conflicting. Always selecting the shortest route towards the base station causes the intermediate nodes to deplete faster, which results in a decreased network lifetime (if the network lifetime is measured by the time that lasts until the first node dies in the entire network).

On the other hand, using a long route composed of many sensor nodes can significantly increase the network delay. At the same time, always choosing the shortest path may result the lowest energy consumption and lowest network delay globally. Ultimately, the routing objectives are tailored by the application; e.g., real-time applications require minimal network delay, while applications performing statistical computations may require maximized network lifetime. Hence, different routing mechanisms have been proposed for different applications [1]. These routing mechanisms primarily differ in terms of network structure and operations they are performing for routing.

1.3.1 Design Constraints for Routing Protocols in Wireless Sensor Networks

The primary design goal of WSNs is to acquire the monitored data from target area and deliver it to base station for its evaluation, while trying to prolong the lifetime of the network. The design of routing protocols in WSNs is influenced by many challenging factors and these factors must be considered to achieve efficient communication in WSNs. Following section, briefly describes some of these design issues that affect routing process in WSNs.

a) Node deployment

Nodes deployment can be deterministic or randomized also known as self-organized and nodes deployment is application specific. In deterministic deployment, sensors are manually placed and data is routed through predetermined paths. Here routes are already known and incase an event occurs, it does not require determining routes thereby saving energy. In random node deployment, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner [2] and if the nodes distribution is not uniform, optimal clustering is employed to allow connectivity and for efficient network operation. Inter-sensor communication is normally within short transmission ranges due to energy and bandwidth limitations. Therefore in WSNs most of the times multi-hop architecture is employed.

b) Energy Efficiency

Routing protocols should prolong network lifetime while maintaining a good grade of connectivity to allow the communication between nodes. It is important to note that the battery replacement in the sensors is infeasible since most of the sensors are randomly placed [3]. Under some circumstances, the sensors are not even reachable. For instance, in wireless underground sensor networks, some devices are buried to make them able to sense the soil [4].So the routing protocols designed must support efficient energy utilization.

c) Data Reporting Model

In applications of WSNs, the data delivery model can be continuous, event-driven, query-driven and even hybrid [24]. In the continuous delivery model, each sensor sends data periodically. In event-driven and query-driven models, the transmission of data is triggered when an event occurs or a query is generated by the sink. Hybrid models are applied in some networks using a combination of continuous, event-driven and query-driven data delivery. The RT routing protocols are highly influenced by the data delivery model, especially with regard to the Minimization of energy consumption and QoS guarantees.

d) Fault Tolerance

Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference. If many nodes fail, MAC and routing protocols must accommodate formation of new links and routes to the data collection base stations. Therefore, multiple levels of redundancy may be needed in a fault-tolerant sensor network.

e) Scalability

The number of sensor nodes deployed in the sensing area may be in the order of hundreds or thousands, or more. Any routing scheme must be able to work with this huge number of sensor nodes.

f) Network Dynamics

Most of the network architectures assume that sensor nodes are stationary. However, mobility of both BS's and sensor nodes is sometimes necessary in many applications [25]. Routing messages from or to moving nodes is more challenging since route stability becomes an important issue, in addition to energy, bandwidth etc. Routing protocols should render appropriate support for these movements.

g) Coverage and Connectivity

A given sensor's view of the environment is limited both in range and in accuracy; it can only cover a limited physical area of the environment. Hence, area coverage is also an important design parameter in WSNs. Sensor nodes are expected to be highly connected to prevent from sensor node failures. And it depends on the, possibly random, distribution of nodes.

h) Data Aggregation

Since sensor nodes may generate significant redundant data, similar packets from multiple nodes can be aggregated so that the number of transmissions is reduced. Data aggregation is the combination of data from different sources according to a certain aggregation function, e.g., duplicate suppression, minima, maxima and average. This technique has been used to achieve energy efficiency and data transfer optimization in a number of routing protocols.

i) Quality of Service

In some applications, data should be delivered within a certain period of time from the moment it is sensed; otherwise the data will be useless. Therefore bounded latency for data delivery is another condition for time-constrained applications. It should be noted that while considering RT support in WSNs, energy efficiency should not be ignored. There is often a tradeoff between these two considerations.

CHAPTER 2

LITERATURE REVIEW

2.1 EVOLUTION OF SENSOR NETWORKS

2.1.1 Application-specific Sensor networks

In application-specific deployments of Sensor networks the application consists of a distributed code that is installed on some or all of the nodes of the network. In simple applications, the same code is installed on all nodes. In more complex applications, different code modules are installed on different nodes. An example of application-specific sensor networks deployments is the wireless sensor network for habitat monitoring deployed in Great Duck Island [7]. In this example, the network is designed for the specific application of habitat monitoring. Based on this application's characteristics, a set of system design requirements were developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. The networks architecture was then built to satisfy those requirements.

Application-specific design has several drawbacks. First, Sensor networks deployed for one or a few applications are often of limited reusability and are, therefore, inherently not cost-effective. This translates into a low return on investment. Another drawback is the tight coupling between the application and the underlying sensor networks. The application is often designed as a monolithic, complex code of tightly coupled modules where each module implements a specific functionality, e.g., user interface, data access (i.e., retrieving sensor readings), actuator activation, etc. To develop these modules, programmers invoke functionalities at several layers in the sensor networks architecture. The reason behind this monolithic, application-specific design is often optimization. By enabling programmers to manipulate parameters and mechanisms at different layers, the code may be tailored to achieve better efficiency for the application at hand. A consequence of this tight coupling between the application and the sensor networks is that considerable reprogramming efforts are often necessary to make the network able to serve new applications.

2.1.2 Generic Sensor networks

Generic Sensor networks are not intended to be used by a specific application. They usually require that a generic code, i.e., the query processing system, be installed on all nodes of the network. Examples of generic query systems include Cougar [8], Tiny DB [9], and REED [10]. This approach also has a number of limitations. First, the same code is installed on each node. A particular node may not need or be able to support all the functionalities of the installed query system. For example, a typical query system would include code for in-network data aggregation, collaborative event detection, actuation coordination, etc. The latter functionality, for example, is not needed at a node with no actuation capabilities. As a result, a sizeable query processing system has to be installed on all nodes of the network regardless of their capabilities. A more important drawback in generic Sensor networks is that they often must trade efficiency for genericity. Typically, a generic query system may not be optimized to fully exploit the specific query patterns of a given application. Also, nodes in a sensor networks do not necessarily have the same hardware configuration.

A generic query system may not be optimized to efficiently manage the hardware resources of specific nodes. As a consequence, generic Sensor networks may not scale to handle high query loads typical in the next-generation, potentially Web-accessible sensor networks. As sensor technologies mature and new applications proliferate, current design models for sensor–actuator systems seem increasingly unable to cope with the requirements of the next-generation of open, ubiquitous, interoperable, multi-purpose sensor networks. Architectures for future sensor systems will have to be able to serve different applications and adapt to different post-deployment query patterns. Networks from different providers will have to be individually programmed and yet able to interoperate efficiently. Both application-specific and generic architectures are obviously unable to satisfy these requirements. To enable the next generation sensor-actuator systems, new customizable architectures are needed.

2.1.3. Customizable sensor networks

Customizable Sensor networks are Sensor networks that are readily configurable, after they are deployed, to serve different types of applications with arbitrary query patterns. A node in a customizable sensor networks would expose its capabilities as identifiable resources that may be accessed by any entity that may communicate with the node and not necessarily by other nodes from the same network. Customizable Sensor networks would provide developers the flexibility to combine the resources provided by nodes in one or different (existing) Sensor networks to meet the requirements of new applications and yet expect the same levels of performance that would result from an application-specific deployment. A possible alternative in building customizable Sensor networks is to use generic Sensor networks as their backbone and to develop additional software layers that customize the functionalities of the generic Sensor networks to satisfy the requirements of the given application. This, however, would only lead to further lower performance and memory availability. Hence Service-Oriented Sensor networks is introduced as a novel approach for building open, interoperable, customizable sensor networks.

2.2 ENERGY EFFICIENT CLUSTERING

Clustering is a process of grouping nodes based on some parameter such as nodes residual energy and node's proximity to its neighbors. Prolonged network lifetime, scalability and load balancing are important requirements for ad-hoc sensor network applications. Cluster sensor nodes are an effective technique for achieving these requirements.

The performance measures of clustering algorithms are

Energy Efficiency

Network Lifetime

Data Accuracy

Latency

- **Energy Efficiency**

A data gathering scheme is energy efficient if it maximizes the functionality of the network. If we assume that all sensors are equally important, we should minimize the energy consumption of each sensor. This idea is captured by the network lifetime which quantifies the energy efficiency of the network.

- **Network Lifetime**

  Network Lifetime is defined as the number of rounds until the first sensor is drained of its energy (Stability Period).

- **Data Accuracy**

  The definition of data accuracy depends on the specific applications for which the sensor network is designed. For instance, in a target localization problem, the estimate of the target location at the sink determines the data accuracy.

- **Latency**

  Latency can be measured as the time delay between the data packets received at the sink and the data generated at the source nodes.

Younis et al. [11] discussed about the significance of clustering and how the energy is saved with clustering techniques. Clustering techniques can aid in reducing useful energy consumption. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Also this technique is needed for load balancing and efficient resource utilization. Clustering can be extremely effective in one-to-many, many-to-one, one-to-any, or one-to-all (broadcast) communication. For example, in many-to-one communication, clustering can support data fusion and reduce data interference

The essential operation in sensor node clustering is to select a set of cluster heads from the set of nodes in the network, and then cluster the remaining nodes with these heads. Cluster heads are responsible for coordination among the nodes within their clusters and aggregation of their data (intracluster coordination), and communication with each other and/or with external observers on behalf of their clusters (intercluster communication).

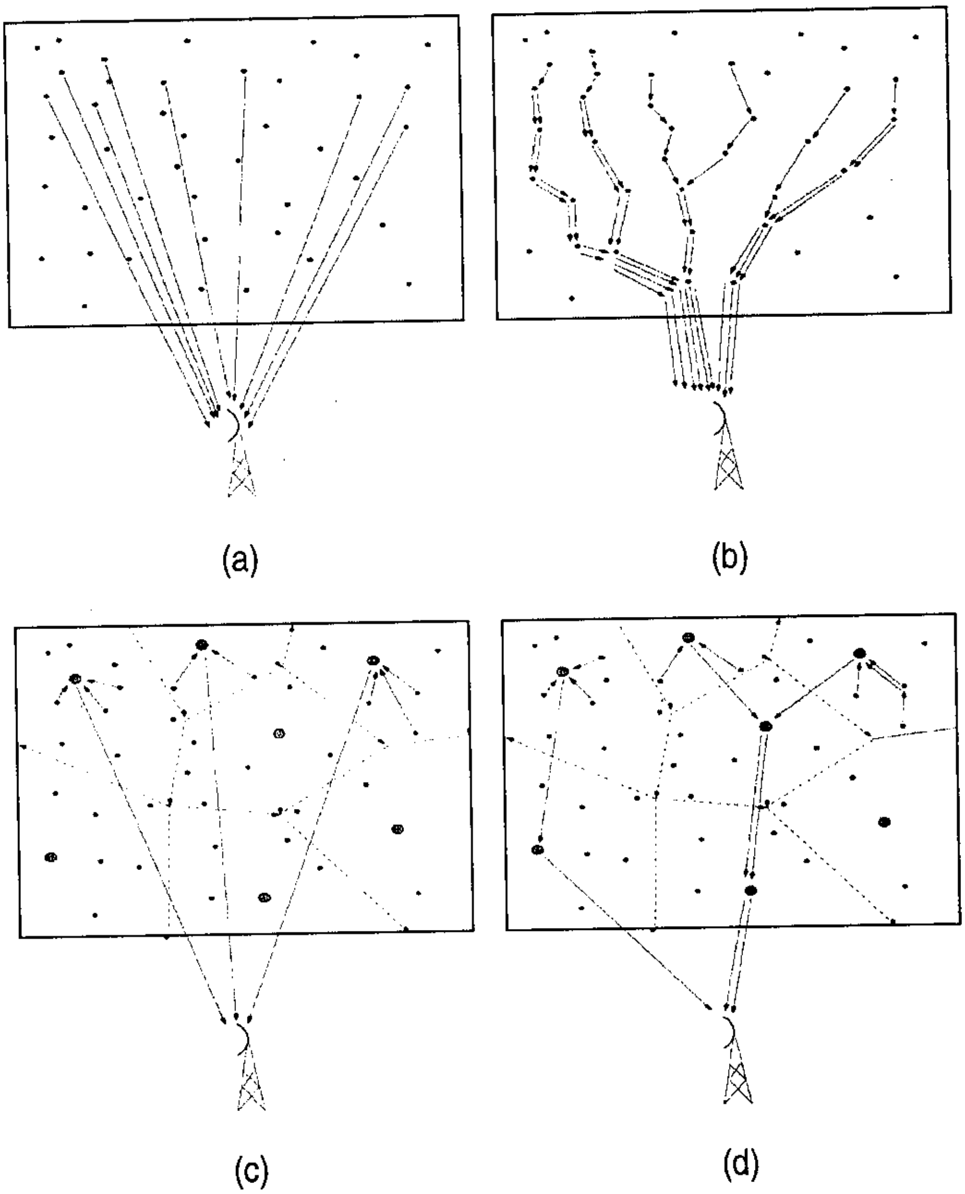


Figure 2.1 Forwarding with and without clustering and aggregation

Figure 2.1. depicts an application where sensors periodically transmit information to a remote observer (e.g., a base station). Figure 2.1-a shows Single hop without clustering in which each sensor node senses the information and that information is transmitted directly to the base station. But in the Single hop with clustering (figure 2.1-c) each non-cluster head nodes send the information to its corresponding cluster head and the cluster head performs the signal processing(eg.data aggregation) on the data and then transmits the data to the base station. The

same process is performed in the multi hop with and without clustering but the data from the cluster head is transmitted through intermediate nodes.

Hence, clustering can reduce the energy consumption and communication overhead for both single-hop and multi-hop networks. Periodic re-clustering can select nodes with higher residual energy to act as cluster heads.

Network lifetime is prolonged through

1. Reducing the number of nodes contending for channel access,
2. Summarizing information and updates at the cluster heads,
3. Routing through an overlay among cluster heads, which has a relatively small network diameter.

There have been substantial amount of research on clustering protocols for WSNs. Most of the clustering protocols utilize two techniques which are selecting cluster-heads with more residual energy and rotating cluster-heads periodically to balance energy consumption of the sensor nodes over the network [12]. These clustering algorithms do not take the location of the base station into consideration. This lack of consideration causes the hot spots problem in multi-hop WSNs. The cluster-heads near the base station die earlier, because they will be in heavier relay traffic than the cluster heads which are relatively far from the base station. To avoid this problem base station distance also considered while forming clusters in ESQR. In order to make wise decisions, it utilizes the residual energy and the distance to the base station parameters of the sensor nodes.

2.3 ROUTING IN WIRELESS SENSOR NETWORKS

2.3.1 Classification of Routing Protocols in WSN

This section describes the survey of state-of-the-art routing protocols for WSNs. In general, routing in WSNs can be classified based on the network structure and protocol operation.

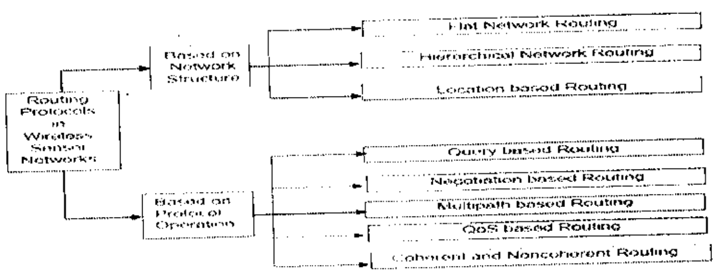


Figure 2.2: Classification of routing protocol

2.3.2 Routing Protocols based on Network-Structure

This section, describes the survey details of the routing protocols that are categorized based on the underlying network structure which can play significant role in the operations of the routing protocol in WSNs.

a) Flat based Routing

In flat network architecture all the sensor nodes are equal and connections between nodes are setup in short distance to establish the radio communication. Route discovery can be carried out in sensor networks using flooding or broadcasting which do not involve topology maintenance. Due to the large number of such nodes, it is not feasible to assign a global identifier to each node. This consideration has led to data centric routing, where the BS sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data.

b) Hierarchy based Routing

Hierarchical routing is the procedure of arranging the routers in a hierarchical manner. In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target. This means that creation of clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the BS. Hierarchical routing is mainly two-layer routing where one layer is used to select cluster heads and the other layer is used for routing.

Flat Routing	Hierarchical Routing
Contention-based scheduling	Reservation-based scheduling
Collision overhead present	Collision avoided
node on multi-hop path aggregates incoming data from neighbors	Data aggregation by cluster head
Links formed on the fly without synchronization	Requires global and local synchronization

Latency in waking up intermediate nodes and setting up the multipath	Lower latency as multiple hops network formed by cluster heads always available
Energy dissipation depends on traffic patterns	Energy dissipation is uniform
Energy dissipation adapts to traffic pattern	Energy dissipation cannot be controlled
Fairness not guaranteed	Fair channel allocation

Table 2.1: Comparison of Flat and Hierarchical routing

c) Location based Routing

In this kind of routing, sensor nodes are addressed by means of their locations. The distance between neighboring nodes can be estimated using incoming signal strengths. Relative coordinates of neighboring nodes can be obtained by exchanging information between neighbors. Alternatively, the location of nodes may be available directly by communicating with a satellite, using GPS (Global Positioning System), if nodes are equipped with a GPS receiver.

To save energy, some nodes are put into sleep mode if there is no activity. More energy savings can be obtained by having as many sleeping nodes in the network as possible.

Classification based on network structure	**Routing protocols**
Flat based	SPIN , Directed Diffusion, Rumor Routing, GBR , MCFA, CADR, COUGAR, ACQUIRE, EAR
Hierarchy based	LEACH, TEEN &APTEEN, PEGASIS, MECN &SMECN, HPAR ,VGA, Sensor aggregate, TTDD
Location-based	GAF, GEAR, SPAN ,MFR, GEDIR, GOAFR

Table 2.2: Routing protocols based on network-structure

2.3.3 Routing Protocols based on Protocol Operation

This section presents the review of routing protocols that has different routing functionality. It should be noted that some of these protocols may fall below one or more of the above routing categories.

a) Multipath routing protocols

The fault tolerance (resilience) of a protocol is measured by the likelihood that an alternate path exists between a source and a destination when the primary path fails. This can be increased by maintaining multiple paths between the source and the destination. These alternate paths are kept alive by sending periodic messages to them. In [13], multipath routing was used to enhance the reliability of WSNs. The proposed scheme is useful for delivering data in unreliable environments. But it increases the traffic significantly. Directed diffusion [14] is a good candidate for robust multipath routing and delivery. Based on the directed diffusion paradigm, a multipath routing scheme that finds several partially disjoint paths is studied in (alternate routes are not node disjoint, i.e., routes are partially overlapped).

b) Query based routing

In this kind of routing, the Base station or sink nodes propagate a query through the network and a node having associated data will send reply to the query initiator. For example, client C1 may submit a query to node N1 and ask: Are there moving vehicles in battle space region 1?. All the nodes have tables consisting of the sensing tasks queries that they receive and send data which matches these tasks when they receive it. Directed diffusion [14] described is an example of this type of routing.

c) Negotiation based routing protocols

These protocols use high level data descriptors in order to eliminate redundant data transmissions through negotiation. The SPIN family protocols are examples of negotiation based routing protocols. The main idea of negotiation based routing is to suppress duplicate information and prevent redundant data from being sent to the next sensor or the base-station by conducting a series of negotiation messages before the real data transmission begins.

d) QoS-based routing

In QoS-based routing protocols, the network has to satisfy certain QoS metrics, e.g., delay, energy, bandwidth, etc. when delivering data to the BS. Sequential Assignment Routing (SAR) proposed in [15] introduces the notion of QoS in the routing decisions. Routing decision in SAR is dependent on three factors: energy resources, QoS on each path, priority level of each packet. In The SAR calculates a weighted QoS metric as the product of the additive QoS metric and a weight coefficient associated with the priority level of the packet. The objective of SAR algorithm is to minimize the average weighted QoS metric throughout the lifetime of the network.

e) Coherent and non-coherent processing

Two examples of data processing techniques pro- posed in WSNs are coherent and non-coherent data processing-based routing [15]. In non-coherent data processing routing, nodes will locally process the raw data before sent to other nodes (aggregators) for further processing. In coherent routing, the data is forwarded to aggregators after minimum processing. The minimum processing typically includes tasks like time stamping, duplicate suppression, etc. To perform energy-efficient routing, coherent processing is normally selected. In [15], a single and multiple winner algorithms were proposed for non-coherent and coherent processing, respectively. The election of a node is based on the energy reserves and computational capability of that node. The Table shows how different routing protocols fit under different category and also compare different routing techniques according to many metrics.

Routing protocols	Mobility	Negotiation based	Data Aggregation	QOS	Multipath	Query
SPIN	Possible	Yes	Yes	No	Yes	Yes
Directed Diffusion	Limited	Yes	Yes	No	Yes	Yes
Rumor Routing	Very Limited	No	Yes	No	No	Yes
GBR	Limited	No	Yes	No	No	Yes
MCFA	No	No	No	No	No	No

CADR	No	No	Yes	No	No	No
COUGAR	No	No	Yes	No	No	Yes
ACQUIRE	Limited	No	Yes	No	No	Yes
EAR	Limited	No	No	No	No	Yes
LEACH	Fixed BS	No	Yes	No	No	No
TEEN & APTEEN	Fixed BS	No	Yes	No	No	No
PEGASIS	Fixed BS	No	No	No	No	No
MECN & SMECN	No	No	No	No	No	No
VGA	No	Yes	Yes	No	Yes	No
Sensor Aggregate	Limited	No	Yes	No	No	Possible
TTDD	Yes	No	No	No	Possible	Possible
GAF	Limited	No	No	No	No	Yes
GEAR	Limited	No	No	No	No	Yes
SPAN	Limited	Yes	No	No	No	Yes
SAR	No	Yes	Yes	Yes	No	No

Table 2.3: Routing protocols based on the operations

2.3.4 Hierarchy based protocols

(i) LEACH protocol

Heinzelman, et. al. [16] introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH randomly selects a few sensor nodes as cluster heads (CHs) and rotates this role to evenly distribute the energy load among the sensors in the network. In LEACH, the cluster head (CH) nodes compress data arriving from nodes and send an aggregated packet to the base station to reduce the amount of information transmitted to the base station. LEACH uses a TDMA/CDMA MAC to reduce inter-cluster and intra-cluster collisions. The operation of LEACH is separated into two phases, the setup phase and the steady state phase. In the setup phase, the clusters are organized and CHs are selected. In the steady state phase, the actual data transfer to the base station takes place. During the setup phase, a predetermined fraction of nodes, p, elect themselves as CHs as follows. A sensor node chooses a random number, r, between 0 and 1. If this random number is less than a threshold value, *T(n)*, the node becomes a cluster-head, and the set of nodes that have not been selected as a cluster-head in the last *(1/P)* rounds, denoted by G. It is given by:

$$T(n) = \frac{p}{1 - p\,(r \bmod (1/p))} \quad \text{if } n \in G$$

Each elected CH broadcast an advertisement message to the rest of the nodes in the network that they are the new cluster-heads. All other nodes, after receiving this advertisement, decide on the cluster to which they want to belong to. Then the long duration steady state phase will continues.

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

It is considered an optimization of the LEACH algorithm. Rather than classifying nodes in clusters, the algorithm forms chains of the sensor nodes. Based on this structure, each node transmits to and receives from only one closest node of its neighbors. With this purpose, the nodes adjust the power of their transmissions [17]. The node performs data aggregation and forwards it the node in the chain that communicates with the sink. In each round, one node in the chain is elected to communicate with the sink. The chain is constructed with a greedy algorithm.

(iii) Threshold-sensitive Energy Efficient Protocols (TEEN and APTEEN)

TEEN [18] is other hierarchical protocol for reactive networks that responds immediately to changes in the relevant parameters. In this protocol a clusters head (CH) sends a hard threshold value and a soft one. The first time a parameter from the attribute set reaches its hard threshold value, the node switches on its transmitter and sends its data. The nodes then transmits data in the current cluster period if the following conditions are true: the current value of the sensed attribute is greater than the hard threshold, and the current value of the sensed attribute differs from sensed value by an amount equal to or greater than the soft threshold. Both strategy looks to reduce energy spend transmitting messages. The main drawback of this scheme is that, if the thresholds are not reached, the nodes will never communicate; the user will not get any data from the network at all and will not come to know even if all the nodes die. Thus, this scheme is not well suited for applications where the user needs to get data on a regular basis.

APTEEN, on the other hand, is a hybrid protocol that changes the threshold values used in the TEEN protocol according to the user needs. In APTEEN, the cluster-heads broadcasts the following parameters: Attributes (A): this is a set of physical parameters which the user is interested in obtaining information. Thresholds: This consists of Hard Threshold (HT) and the Soft Threshold (ST).Schedule: this is a TDMA schedule, assigning a slot to each node. Count Time (CT): it is the maximum time period between two successive reports sent by a node. Once a node senses a value beyond HT, it transmits data only when the value of that attributes changes by an amount equal to or greater than the ST. If a node does not send data for a time period equal to the count time, it is forced to sense and retransmit the data. APTEEN uses a modified TDMA schedule to implement the hybrid network. The main features of the APTEEN scheme include the following. It combines both proactive and reactive policies. It offers a lot of flexibility by allowing the user to set the count-time interval (CT).The main drawback of the scheme is the additional complexity required to implement the threshold functions and the count time.

2.3.5 Location based protocols

(i) Geographic Adaptive Fidelity (GAF)

In [23][26] the authors proposes GAF - an energy-aware location-based routing algorithm. The network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS. Each node uses its GPS-indicated location to associate itself with a point in the virtual grid. There are three states defined in GAF. These states are discovery, for determining the neighbors in the grid, active reflecting participation in routing and sleep when the radio is turned off. In order to handle the mobility, each node in the grid estimates it's leaving time of grid and sends this to its neighbors. Before the leaving time of the active node expires, sleeping nodes wake up and one of them becomes active.

Then, cluster head is responsible for receiving raw data from other nodes in its cluster and forward it to the BS. GAF strives to keep the network connected by keeping a representative node always in active mode for each region on its virtual grid. Simulation results show that GAF performs at least as well as a normal ad hoc routing protocol in terms of latency and packet loss and increases the lifetime of the network by saving energy.

(ii) Geographic and Energy Aware Routing (GEAR)

The protocol, called Geographic and Energy Aware Routing (GEAR), uses energy aware and geographically-informed neighbor selection heuristics to route a packet towards the destination region. The key idea is to restrict the number of interests in directed diffusion by only considering a certain region rather than sending to the whole network. By doing this, GEAR can conserve more energy than directed diffusion. Each node in GEAR keeps an estimated cost and a learning cost of reaching the destination through its neighbors. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a refinement of the estimated cost that accounts for routing around holes in the network. A hole occurs when a node does not have any closer neighbor to the target region. If there are no holes, the estimated cost is equal to the learned cost. The learned cost is propagated one hop back every time a packet reaches the destination so that route setup for next packet will be adjusted. There are two phases in the algorithm: (1) forwarding packets towards the target region: Upon receiving a packet, a node checks its neighbors to see if there is one neighbor, which is closer to the target region than itself. If there is more than one, the nearest neighbor to the target region is selected as the next hop. If they are all further than the node itself, this means there is a hole. In this case, one of the neighbors is picked to forward the packet based on the learning cost function and (2) forwarding the packets within the region: If the packet has reached the region, it can be diffused in that region by either recursive geographic forwarding or restricted flooding. Restricted flooding is good when the sensors are not densely deployed. In high-density networks, recursive geographic flooding is used. In that case, the region is divided into four sub regions and four copies of the packet are created. This splitting and forwarding process continues until the regions with only one node are left.

(iii) SPAN:

Another position based algorithm called SPAN [19] selects some nodes as coordinators based on their positions. The coordinators form a network backbone that is used to forward messages. A node should become a coordinator if two neighbors of a non-coordinator node cannot reach each other directly or via one or two coordinators (3 hop reachability). New and existing coordinators are not necessarily neighbors in which makes the design less energy efficient because of the need to maintain the positions of two or three hop neighbors in the complicated SPAN algorithm.

2.3.6 Drawbacks of Existing Routing Techniques

Most existing query routing approaches are "overreaching", i.e., they use broadcasting schemes where nodes use some default transmission power to broadcast queries to all of their neighbors. Moreover, current routing schemes generally do not exploit the semantics of queries. As a result, a query may be routed through several hops to end up at nodes that do not provide the sensing capabilities that it requires or that do not have paths to nodes that provide the required capabilities. This obviously incurs excessive communication and processing overhead that is not strictly required to deliver the query to its recipients.

2.4 SERVICE-ORIENTED QUERY ROUTING IN WIRELESS SENSOR NETWORKS:

2.4.1 Need for Service Oriented Query Routing

To address the deficiencies of existing routing protocols, the notion called service-driven query routing (SQR) is introduced which is a routing scheme based on the idea of modeling nodes' capabilities as services. Sensor network nodes have heterogeneous sensing capabilities. Each node exposes its capabilities as services. Conceptually, a service is a computational component that: (i) has a unique network-wide identifier, (ii) may be invoked asynchronously, (iii) may have one or more parameters, and (iv) produces one or more values as a result of the invocation. A service may have multiple service instances, each running on a given node. The Service-Oriented Sensor Networks has one or more base stations. Users query the network by submitting queries to one of its base stations or directly to individual nodes.

This works by forming clusters of nodes based on various parameters which are also used to form the queries. In this approach hierarchy of clusters are formed and each cluster has its own cluster head which are used to route the queries to reach the specific nodes which is providing requested service

2.4.2 Service Oriented Query Routing

Sensor network nodes have heterogeneous sensing and actuation capabilities. The sensor network has one or more base stations. Users query the sensor network by submitting queries to one of its base stations. Queries may be of two types: task-driven and event-driven queries:

Query :: event <event>;

condition <condition>;

action <action>;

space <spatial scope specification>;

time <temporal scope specification>

- Task-driven Queries: In a task-driven query, the application requests a reading to be retrieved from one or more sensors or an operation to be performed by one or more actuators. The result of a task-driven query is the value of the reading or a code indicating the outcome of the actuation operation.
- Event-driven Queries: In an event-driven query, the application requests to be notified when an event of interest occurs. Typically, the result of such an event-driven query is a notifier message that informs the application of the occurrence of the event. A notifier may include additional information such as the time when the event occurred, the geographical location where the event occurred, etc.

Both task-driven and event-driven queries may be one-time queries or repetitive queries. A repetitive query is a query that an application submits to request that a sensing/actuation task be carried out or that an event be detected repetitively with a

Given frequency and for a given duration.

In this approach extended ECA (Event-Condition-Action) model for query specification is adapted. In general, queries specify five elements: an event, a condition, an action, a spatial scope, and a temporal scope. Here the acronym ECAST (Event-Condition-Action-Spatial scope-Temporal scope) is used to refer to the query model. A query in the ECAST query model has the following syntax:

Where:

• <event> is the event that triggers the execution of the query's action. An event is a condition expressed in terms of attributes that may be sensed. For example, event temp > 70 or light > 110 is an event that occurs when one of the given conditions becomes valid.

• <condition> is a condition that must be satisfied to execute the query's action. Note that, in queries that specify both an event and a condition, the condition is evaluated only when the event occurs; the event condition must be evaluated continuously to detect the occurrence of the event. The query's condition is also specified in terms of attributes. For example: condition temp >65.

• <action> is an invocation of a service. Each service invocation specifies the service identifier and the values for the service parameters. For example: action getLight() requests that nodes involved in the query read and return the value of their light sensors.

• <spatial scope specification> specifies the geographical area where the sensing/actuation is to take place. For example, space Circle (ni,10ft) specifies all the nodes within 10 feet from node ni.

• <temporal scope specification> specifies the start and end times for the query and, for repetitive queries, the query's frequency. The start (resp. end) time is the time at (or after) which the user wants the query execution to begin (resp. end). The frequency specifies the time that must elapse between two successive executions of the query. If the frequency is not specified, a new iteration of the query is started immediately upon the completion of the current iteration.

For example,

Q1 :: event temp > 65 ;
condition humidity > 50%;
action getLight();
space Circle(n5, 10 ft);
time start 10:00am

end 5:00pm
frequency 100s;

This query requires all nodes within a 10-feet distance from node n5 to repetitively read their light sensor when their temperature reading exceeds 65 if the condition humidity > 50% is true. The query also specifies a start time of 10:00am and an end time of 5:00pm as well as a frequency of 100s.

2.4.3 μRACER: A Reliable Adaptive Service-Driven Efficient Routing Protocol suite for Sensor Networks

In [20] the author proposed a routing protocol suite based on a novel service-oriented design for sensor networks where nodes expose their capabilities to applications as a service profile. A node's service profile consists of a set of services (i.e., sensing and actuation capabilities) that it provides and the quality-of-service (QoS) parameters associated with those services (delay, accuracy, freshness, etc.). μRACER uses an efficient service-aware routing approach that aggressively reduces downstream traffic (from the sink to the network's nodes) by translating service profiles into efficient paths for queries. To support QoS, μRACER dynamically adapts each node's routing behavior and service profile according to the current context of that node, i.e., number of pending queries and number and type of messages to be routed. Finally, μRACER improves end-to-end reliability through a scalable reputation-based approach in which each node is able to locally estimate the next hop of the most reliable path to the sink. Service- and context-aware reliable routing enhances the network's efficiency and effectiveness (satisfaction of applications' QoS requirements).μRACER was implemented on top of TinyOS and conducted several experiments that confirmed μRACER's ability with regard to each of its design objectives.

FEATURES OF μRACER:

μRACER is developed to provide four features not simultaneously available in any previous routing protocol:

1. Exploiting message semantics and node capabilities,

2. Context-aware task scheduling,

3. Local and autonomous determination of communication context, and

4. Direction-aware routing.

Architecture of μRACER:

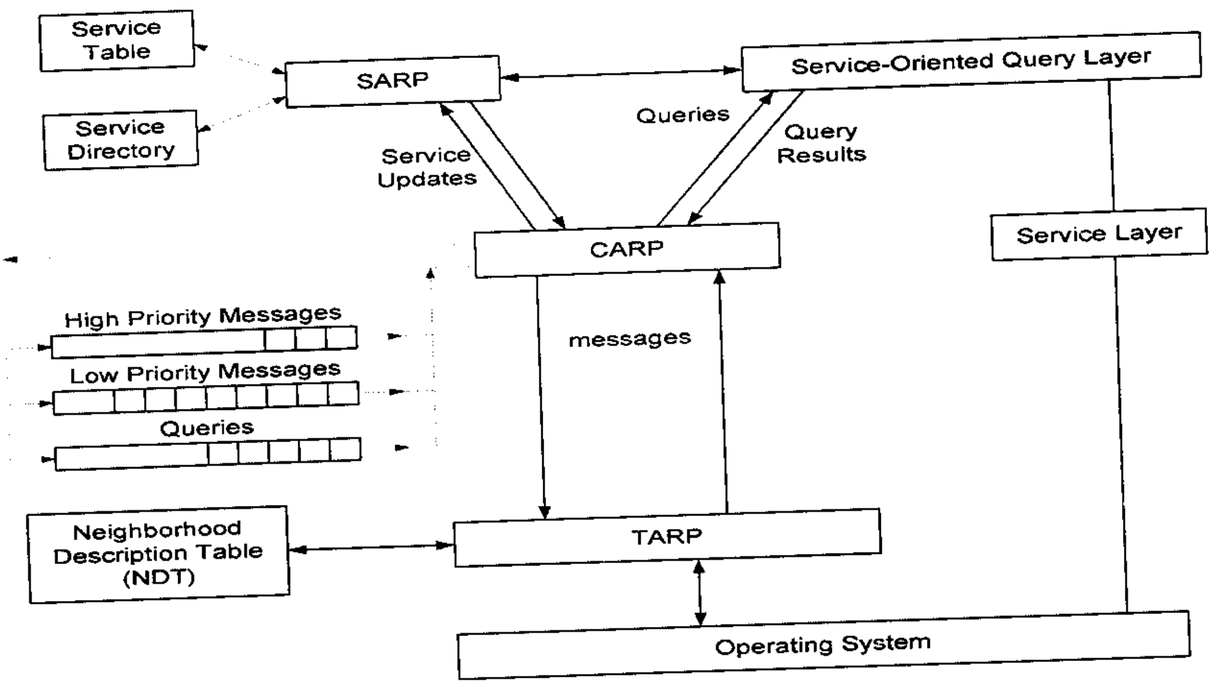


Figure 2.3: Architecture of μRACER

Figure 8 shows architecture of μRACER. It consists of three routing protocols, namely, TARP, CARP, and SARP. TARP monitors nodes' behavior and links' quality and provides a routing layer that supports high end-to-end reliability. CARP and SARP are built on top of TARP. CARP receives messages (service updates, queries, and query results) from TARP. It forwards service updates to SARP and queries to the SOQL (Service-Oriented Query Layer). CARP also receives (outgoing) service updates from SARP and query results from the SOQL. It forwards these messages to TARP which is responsible for actually sending them to other neighbors. CARP is central to a node's operation. It determines the best activity (processing queries, sending messages, sleeping) for a node at any given time. CARP also provides priority-based routing that enables urgent query results to reach their destination faster than regular messages. SARP routes queries from their source (typically, the network's sink) to their destinations based on service reachability. It uses an aggressive path pruning approach whereby nodes discard messages if no path is known to nodes that provide the service requested in a query message. Through its three protocols, μRACER delivers μan integrated approach to routing that provides energy efficiency, reliability, scalability, and support for QoS.

2.4.4 Service Driven Query Routing (SQR)

In traditional query routing approaches the query is disseminated through the entire network. Nodes receive and forward query messages to other nodes without regard to whether forwarding a given query to a given neighbor would actually contribute to answering the query. To reduce the cost of query routing, In [21] the authors introduces the idea of service-driven query routing (SQR) which is a capability-based routing scheme. The basic idea of SQR is to exploit the semantics of a query to reduce the number of nodes that receive the query. Each node that receives a query Q broadcasts it only within a minimal transmission range that includes the neighbors that provide the service requested by the query Q or are on a path to a node that provides that service. An assumption in our approach is that nodes are capable of variable-range transmission. This has recently become a reasonable assumption with the emergence of RF transceiver with programmable output power (e.g., ChipCon C1000 RF transceiver.) Also, several studies have indicated that variable range transmission approaches for routing usually outperform the common-range transmission approach in terms of power savings and increased capacity. Two variants are proposed for this approach. In the first variant, called basic service driven query routing, nodes use static information initialized at bootstrapping time to make decisions about query routing. The second variant is an extended version of the first one where nodes also use dynamic information about network connectivity to further improve their query routing decisions.

This approach is based on two key ideas: enabling nodes to be aware of which services are available on which paths and enabling nodes to compute local optimal paths to reach nodes providing a given service. To support this two requirements, two concepts are introduced that translate into two data structures maintained by each node: (i) service bitmaps and (ii) transmission power graphs

In contrast with current Sensor Networks, nodes in service-oriented Sensor Networks expose their capabilities in the form of services. This enables new alternatives to improve query routing and processing.

2.4.5 Service Aware Routing Protocol (SARP)

Abdelmounaam Rezgui and Mohamed Eltoweissy [22, 21] developed a service aware routing protocol (SARP) that routes a query based on its semantics from one or more base stations to any node that can provide a response to the query.

The key idea in SARP is avoiding flooding the network with the query as this may result in "aimlessly routing" the query to nodes that neither provide the requested service nor are on a path to nodes that provide the requested service.

The nodes' sensing capabilities (temperature, lighting, humidity and so forth) are exposed to applications in the form of a collection of programmatic abstractions called services.

SARP routes queries as follows. First the nodes advertise their sensing and actuation services to their neighbors at bootstrapping time. Any node in the neighborhood learning of the provision of a new service that it doesn't provide further disseminates the advertisement to its neighbors. By the time this first phase of SARP is over, each node will have a service directory indicating which node provides what service or which node is on a path to a node that provides a specific service.

When a query is received by a node from a base station the node checks the service requested in the query. If the service can be provided by that node, a response will be provided. Otherwise, the node will have to broadcast the query to its neighbors after checking its service directory to see if at least one of its neighbors can either provide the service or is on a path to a node that provides the service.

SARP is a novel approach to provide content based routing of queries in a WSN. However, SARP has its drawbacks that need to be tackled and which are to set out to solve in preparing this paper.

Two of SARP's drawbacks arise from the fact that it is a flat based routing mechanism and not a hierarchical one. In a flat based routing approach such as SARP the nodes are organized linearly and data has to be transmitted from the base station to a node by passing through all intermediate nodes between the source and the sink. In such a scheme, the nodes close to the base station will die out quickly as these nodes will be repeatedly used and disproportionately exhausted [23]. In addition, a linear organization of the network topology may not be the most energy efficient as explained in [23].

Another drawback of SARP is that it duplicates routing state on each node. While distributing the routing state as much as possible has got advantages in terms of robustness, the fact that all nodes need to store the routing state means a higher use of node memory in the WSN overall. In addition, updates of routing state change will require higher maintenance traffic.

2.4.6 Hierarchical Service-Oriented Query Routing

Yazhy and Park[6] Developed a data-centric query routing protocol called HESOR (HiErarchical Service Oriented Routing of queries) that not only routes queries from base stations to WSN nodes based on the content of the requested data but also tries to minimize the energy expended by the nodes in so doing. HESOR also distributes the routing state in a hierarchy of cluster heads instead of centralizing routing state on one or two base stations.

The envisioned scenario is one in which there are heterogeneous nodes that sense various data, store the data locally and respond to user queries submitted through one or more base stations. It is also assumed that these nodes are capable of using variable transmission power.

A major modification of HESOR algorithm from that used in LEACH in forming the clusters is that of using the service bitmap as a criterion in addition to signal strength to organize nodes into clusters of services. In LEACH, the nodes belong to any cluster head in their vicinity with the strongest radio signal. In HESOR, the nodes belong to a service cluster head in their vicinity (with the strongest signal strength) that provides the same service as they do. This allows for the possibility of providing an aggregate data as a response to a query from the base station as all temperature sensing nodes report to the temperature service cluster head, humidity sensing nodes report to the humidity service cluster head and so forth.

Note here that the regional cluster heads have to always be on while the individual nodes and the service cluster heads go to sleep periodically based on the TDMA schedule assigned to them from their cluster heads respectively.

In addition, when the nodes in one region communicate with their service cluster head or the service cluster heads communicate with their regional cluster head, interference may occur with the communication in neighboring regions. In order to avoid such kind of interference in the transmission of data between the various nodes and service cluster heads, nodes in each region communicate using different CDMA codes assigned by the regional cluster head. However, the regional cluster heads all use the same CDMA code to allow communication between them.

CHAPTER 3

METHODOLOGY

3.1 ENERGY EFFICIENT SERVICE ORIENTED ROUTING OF QUERIES

Energy consumption is the most important factor to determine the life of a sensor network because usually sensor nodes are driven by battery and have very low energy resources. This makes energy optimization more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. This can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes and the entire network and not only in the individual nodes. A sensor node usually consists of four sub systems that are computing subsystem, Communication subsystem, Sensing subsystem, Power supply subsystem. Analysis shows that communication module consumes most of the sensor energy. Since routing protocol directly access the communication module it needs to be designed as Energy Efficient.

Before describing the proposed algorithm in detail, the characteristics of the system model is introduced, this is used for implementation. First, the assumptions are listed for the network model.

1. Sensor nodes are deployed randomly
2. All sensor nodes and the base station are stationary after deployment phase
3. Nodes have the capability of adjusting the transmission power according to the distance of the receiver nodes
4. All sensor nodes have the same amount of energy when they are initially deployed
5. Base station need not be located far away from the sensing region
6. All sensor nodes are identical and knows the distance from the base station.

Previously proposed query routing schemes are either not content based or lack to take advantage of a hierarchical organization of the network when they are content based. ESQR uses content based clusters to hierarchically organize the network. The clusters are formed not just based on energy efficiency but also on similarity of services provided by nodes belonging to the same cluster. This will help in providing an aggregate response to the query if necessary. ESQR has two components which are described below.

3.1.1 Cluster Formation

In order to gather information more efficiently, wireless sensor networks (WSNs) are partitioned into clusters. Clustering helps the nodes to minimize the overall energy dissipation in the network by allowing only some nodes to take part in the transmission to the base station. Moreover it also helps to reuse the bandwidth and thus utilizes better resource allocation and improved power control.

In general, each cluster has a cluster-head which coordinates the data gathering and aggregation process in a particular cluster. Clustering in WSNs guarantees basic performance achievement with a large number of sensor nodes. In other words, clustering improves the scalability of WSNs. This is because clustering minimizes the need for central organization and promotes local decisions. Cluster-heads are rotated in each round. The term round refers to the interval between two consecutive cluster formation process.

There have been substantial amount of research on clustering protocols for WSNs. Most of the clustering protocols utilize two techniques which are selecting cluster-heads with more residual energy and rotating cluster-heads periodically to balance energy consumption of the sensor nodes over the network. These clustering algorithms do not take the location of the base station into consideration. This lack of consideration causes the hot spots problem in multi-hop WSNs. The cluster-heads near the base station die earlier, because they will be in heavier relay traffic than the cluster heads which are relatively far from the base station. In order to avoid this problem the network is partitioned into clusters with different sizes using the term called as competition radius. The clusters close to the base station are smaller than the clusters that are far from the base station.

In fact considering only one factor, like energy, is not suitable to elect the cluster-head properly. This is because other conditions like distance of the nodes with respect to the entire cluster, also gives a measure of the energy dissipation during transmission for all nodes. The more central the node is to a cluster, the more is the energy efficiency for other nodes to transmit through that selected node. The concentration of the nodes in a given region also affects in some way for proper cluster-head election. It is more reasonable to select a cluster-head in a region, where the node energy is high. In this proposed algorithm both residual energy and distance with base station is considered using fuzzy logic to form the clusters.

Fuzzy logic is based on the idea that all things admit of degrees. It attempts to model our Sense of words, our decision making and our common sense. Unlike Boolean logic having two values, fuzzy logic is multi-valued and uses continuum of logical values or degrees of membership between 0 and 1.The basic idea of the fuzzy set theory is that an element belongs to a fuzzy set with a certain degree of membership. This degree is usually taken as a real number in the interval [0, 1]. For a given fuzzy set, the x-axis represents the universe of discourse – the range of all possible values applicable to a chosen variable and the y-axis represents the membership value of the fuzzy set.

The characteristic function of a crisp set A, can be defined as $f_A(x)$, where X is the universe of discourse with its elements denoted as x

$$f_A(x) = \begin{cases} 1, \text{if } x \in A \\ 0, \text{if } x \notin A \end{cases}$$

$$f_A(x): X \rightarrow \{0, 1\},$$

Hence, for any element x of universe X, characteristic function $fA(x)$ is equal to 1 if x is an element of set A, and is equal to 0 if x is not an element of A.

Fuzzy Logic Control

The model of fuzzy logic control consists of a fuzzifier, fuzzy rules, fuzzy inference engine, and a defuzzifier.. The process is performed in four steps:

- Fuzzification of the input variables energy, concentration and centrality - taking the crisp inputs from each of these and determining the degree to which these inputs belong to each of the appropriate fuzzy sets.

- Rule evaluation - taking the fuzzified inputs, and applying them to the antecedents of the fuzzy rules. It is then applied to the consequent membership function (Table 3.1).

- Aggregation of the rule outputs - the process of unification of the outputs of all rules.

- Defuzzification - the input for the defuzzification process is the aggregate output fuzzy set chance and the output is a single crisp number. During defuzzification, it finds the point where a vertical line would slice the aggregate set chance into two equal masses

Fuzzy Rules and Fuzzy Sets

A conditional statement based on two linguistic values A and B on universe of discourse X and Y with linguistic variables x and y of the form

IF x is A THEN y is B, defines a fuzzy rule.

In fuzzy expert systems, linguistic variables are used in fuzzy rules which relates to fuzzy sets. For example:

IF energy is high
THEN cluster-head chance is high

The range of possible values of a linguistic variable represents the universe of discourse of that variable. To site an example, the universe of discourse of the linguistic variable energy may have the scaled range between 0 and 100, and can include fuzzy sets as low, medium, and high. All rules fire to some extent, depending upon the degree of membership to which the antecedent relates with the consequent. A simple example may be,

IF the energy is low and
IF the concentration is low and
IF the centrality is far
THEN the node's cluster-head election chance is very small.

Expert Knowledge Representation

Expert knowledge is represented based on the following three descriptors:

- Node Energy - energy level available in each node, designated by the fuzzy variable energy,
- Distance to Base - a value which specifies the distance with base station for a particular node.

To find the node distance, the base station selects each node and calculates the sum of the squared distances of other nodes from the selected node. Since transmission energy is proportional to d^2, the lower the value of the distance, the lower the amount of energy required by the other nodes to transmit the data through that node as cluster-head.

The linguistic variables used to represent the node energy are divided into three levels: low, medium and high, respectively, and there are three levels to represent the node distance: close, medium and far, respectively. The outcome to represent the node cluster-head election chance was divided into nine levels: very small, small, rather small, medium small, medium, medium large, rather large, large, and very large. The fuzzy rule base currently includes rules like the following: if the energy is high and the distance is far then the node's transmission range is very large.

So 3*3 = 9 rules for the fuzzy rule base.

Distance to Base	Residual Energy	Competition Radius
Close	Low	Very Small
Close	Medium	Small
Close	High	Rather Small
Medium	Low	Medium Small
Medium	Medium	Medium
Medium	High	Medium Large
Far	Low	Rather Large
Far	Medium	Large
Far	High	Very Large

Table 3.1: COMPETITION RADIUS CALCULATION

The first one is the distance to the base station of a particular tentative cluster-head. The fuzzy set that describes the distance to base the station input variable is depicted in Figure 1. The linguistic variables for this fuzzy set are close, medium and far. A trapezoidal membership function is chosen for close and far. On the other hand, the membership function of medium is a triangular membership function.

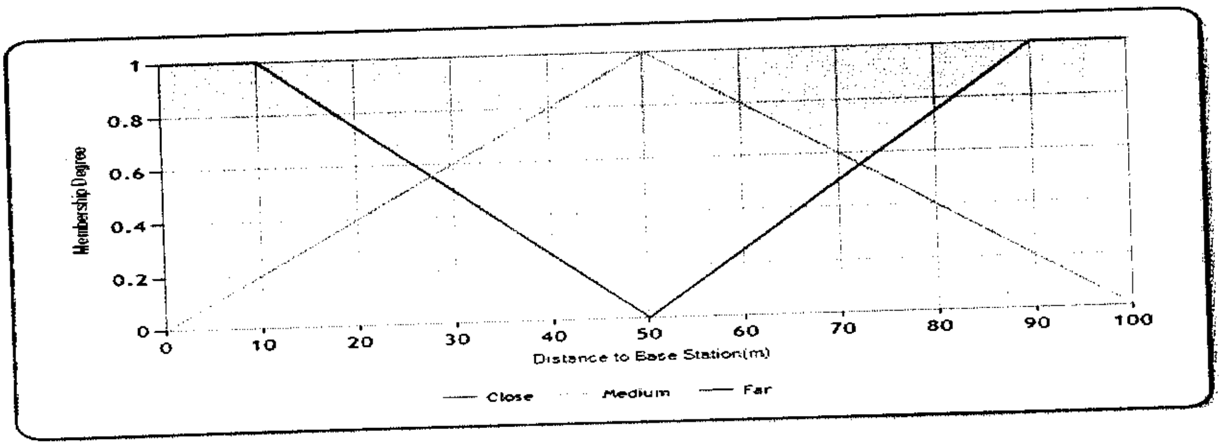


Fig. 3.1 Fuzzy set for fuzzy input variable DistanceToBase

The second fuzzy input variable is residual energy of the tentative cluster-head. The fuzzy set that describes residual energy input variable is illustrated in Figure 2. Low, medium and high are the linguistic variables of this fuzzy set. Low and high linguistic variables have a trapezoidal membership function while medium has a triangular membership function.

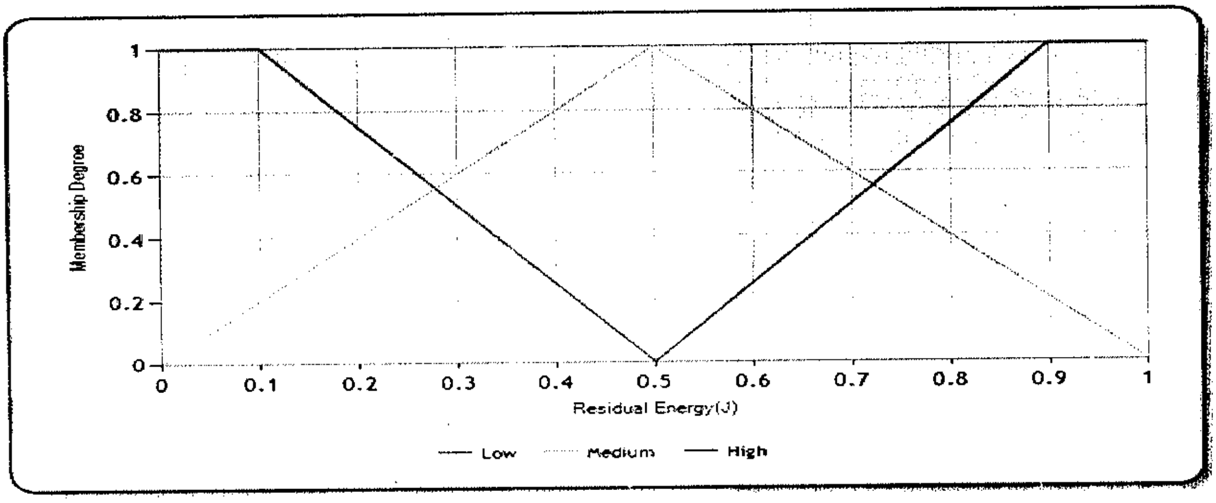


Fig. 3.2 Fuzzy set for fuzzy input variable ResidualEnergy

The only fuzzy output variable is the competition radius of the tentative cluster-head. Fuzzy set for competition radius fuzzy output variable is demonstrated in Figure 3.Here there are 9 linguistic variables which are very small, small, rather small, medium small, medium, medium large, rather

large, large and very large. very small and very large have a trapezoidal membership function. The remaining linguistic variables are represented by using triangular membership functions.

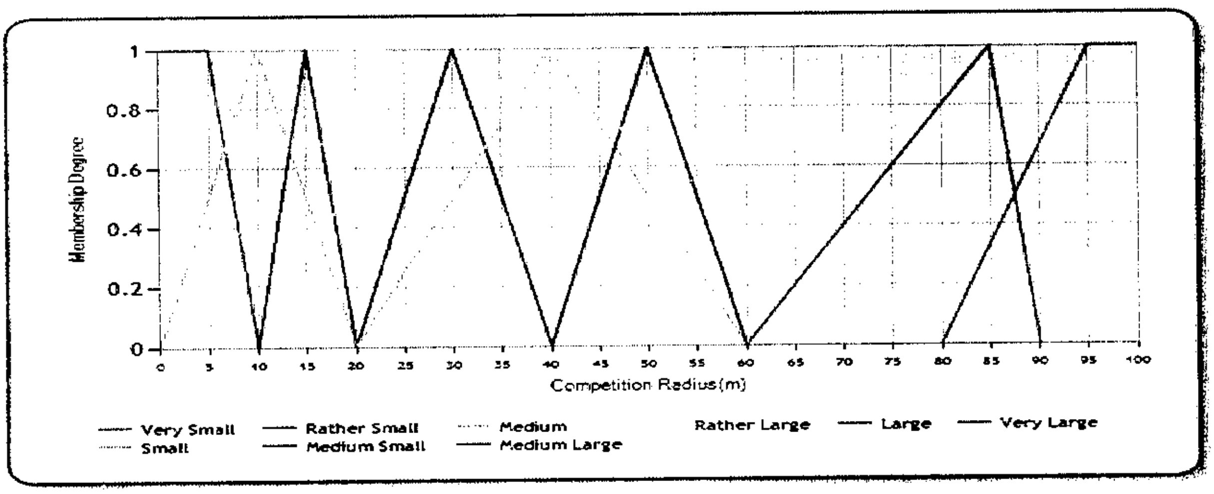


Fig. 3.3 Fuzzy set for fuzzy output variable Competition Radius

Fig 3.4 shows ESQR's cluster formation algorithm that is executed at bootstrapping time and periodically in predetermined time intervals on each node.

ALGORITHM: Cluster Formation

Input: Sensor network with randomly deployed nodes

Output: Service clusters

```
nodeState ← CLUSTERMEMBER
clusterMembers ← empty
myClusterHead ← this
be TentativeHead ← TRUE
Calculate Rcomp using fuzzy if-then mapping
message(ID,Rcomp,resEnergy)
On receiving Message from node N
if this.resEnergy < N.resEnergy then
  be TentativeHead ← FALSE
  Advertise QuitElectionMessage(ID)
end if
if be TentativeHead = TRUE then
  AdvertiseCandidateCHMessage(ID,
  service, resEne rgy,distance)
  nodeState← CLUSTERHEAD
end if
On receiving all candidateCHMessage
  For each message
    If this.service is same as received service
    myClusterHead ← the closest cluster- head
end if
Send JoinCHMessage to the closest clusterhead
On receiving JoinCHMessage(ID) from node N
add N to the clusterMembers list
EXIT
```

Fig: 3.4 CLUSTER FORMATION ALGORITHM

In Fig 3.4, Rcomp and resEnergy represents the competition radius and the residual energy of the sensor node, respectively. Each node in the network calculates the competition radius based on two parameters that are distance with base station and residual energy. The fuzzy if-then mapping rules were used to find the competition radius.

Fuzzy logic systems can manipulate the linguistic rules in a natural way and are capable of making real time decisions, even with incomplete information. Simulation shows that depending upon network configuration a substantial increase in network lifetime can be accomplished as compared to probabilistically selecting the nodes as cluster-heads using only local information. For a cluster, the node elected by the base station is the node having the maximum chance to become cluster-head using three fuzzy descriptors - energy level in each node and node distance with base station, and minimizing energy consumption for all nodes consequently increasing the lifetime of the network.

There are nine combinations of competition radius (ie transmission range) will be created based on three categorization of both distance and residual energy as shown in table 1.Nodes in the network has the capability of adjusting its transmission range based on the calculated competition radius.

After each node determines its competition radius, cluster-head competition begins. Each node broadcast its residual energy along with its ID using calculated Rcomp. Each receiving node compares its residual energy with received residual energy. If its residual energy is lesser than received residual energy then the node will advertise Quit Election Message. Otherwise the node will become a tentative cluster-head. Each tentative cluster-head advertises CandidateCHMessage to compete with other tentative cluster-heads locally. This message is advertised to the tentative cluster-heads which are inside the maximum cluster-head competition radius. It includes node ID, residual energy level of the source node, Distance to the base station and the service provided by the node. Residual energy is the key parameter in cluster-head competition.

The nodes receiving CandidateCH message will check for the service match and if more than one nodes provides same service it will selects the cluster-head which is more closer to the base station.

3.1.2 Query Routing

Query processing in sensor networks entails three phases: (i) query routing, (ii) query evaluation, and (iii) aggregating results and routing them back to the query source. Most existing query routing approaches are"overreaching", i.e., they use broadcasting schemes where nodes use some default transmission power to broadcast queries to all of their neighbors. Moreover, current routing schemes generally do not exploit the semantics of queries. As a result, a query may be routed through several hops to end up at nodes that do not provide the sensing capabilities that it requires or that do not have paths to nodes that provide the required capabilities. This obviously incurs excessive communication and processing overhead that is not strictly required to deliver the query to its recipients.

Query routing in ESQR is a distributed process in which several nodes cooperate in routing queries requesting services toward nodes providing those services. Typically, a query is initiated by a base station and it requests the invocation of a given service by a given subset of nodes. Let Q(s) be a query that the base station issues requesting the invocation of service s.
In this first order radio model is used, to model the energy benefits of using a clustered service oriented query routing approach such as ESQR over a linear query routing approach. To transmit a k-bit message a distance d using this radio model, the energy expended by the radio is specified as:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d)$$
$$E_{Tx}(k,d) = E_{elec} * k + \epsilon_{amp} * k * d^2$$

The energy required to receive message is also needs to be considered (query):

$$E_{Rx}(k) = E_{Rx-elec}(k)$$
$$E_{Rx}(k) = E_{elec} * k$$

To transmit a message, energy expenditure consists of two elements, that of the energy for the transmit electronics $(E_{Tx-elec})$ and for the transmit amplifier (ϵ_{amp}) whereas the energy expended for receiving is represented by $(E_{Rx-elec})$.

The primary design goal of WSNs is to acquire the monitored data from target area and deliver it to base station for its evaluation, while trying to prolong the lifetime of the network.

Fig 3.2 shows the algorithm for query routing that routes a query based on its semantics from one or more base stations to any node that can provide a response to the query. The nodes' sensing capabilities (temperature, lighting, humidity and so forth) are exposed to applications in the form of a collection of programmatic abstractions called services.

A key idea of EQSR is that of forming hierarchical clusters based on parameters that will ultimately be used in specifying a user query. Thus, here first organize the nodes into service clusters and the service clusters into regional clusters

ALGORITHM: Query Routing

Input: specifying the query with required service

Output: Query routed to the node which is providing requested service

1. When a query is received at regional cluster head
    a. Examine query to determine which region the query belongs to.
    b. IF query belongs to same region as the receiving regional cluster head
        i. Examine the service requested by the query
        ii. If itself can provide the service then routes the query
    c. ELSE (query belongs to another region)
        i. Compute a waiting time t=1/RSSI of the concerned regional cluster head.
        ii. Listen for the rebroadcast of this same query by other regional cluster heads until 't' expires
        iii. IF query is broadcast by another regional cluster head before the expiration of time t DO NOTHING
        iv. ELSE broadcast query upon the expiration of t
2. When query is received at service cluster head:
    a. IF aggregate is required respond directly to regional cluster head using maximum transmission power level

b. ELSE broadcast query using maximum transmission power level so that it is received by any one node providing that service currently not sleeping.

Fig 3.5: QUERY ROUTING ALGORITHM

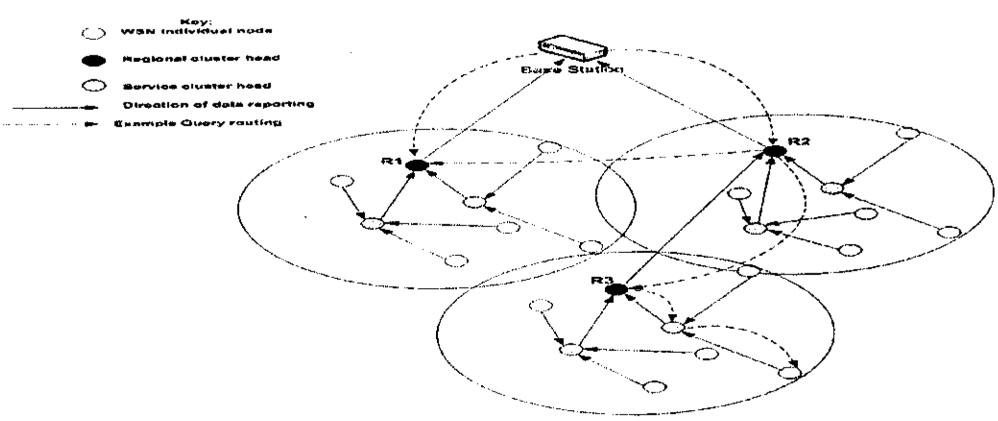


Fig 3.6: CLUSTER FORMATION SCENARIO

In the example scenario for Fig. 3.3, the query is destined to region 4. Hence, once the query reached R1 and R2, which one of regional cluster heads R1 or R2 should further broadcast the query is determined by our algorithm (Fig 3.2) based on a waiting time computed as a reciprocal (multiplicative inverse) of the received signal strength indicator (RSSI) from R3. The RSSI figure of R3 at R1 and R2 was established during cluster formation time.

In our example in Fig. 1, the RSSI from R3 was higher at R2 than at R1. Hence, R2 – not R1- broadcasted the query since R1 had to wait a longer time based on the fore mentioned computation (i.e. 1/RSSI) and thus learned while waiting that the query had already been broadcasted by R2 earlier.

When the query reaches a service cluster head, It is broadcasted further to a specific node based on the destination node's information. Alternatively, the service cluster head can respond directly with an aggregated data by reporting the data back to the regional cluster head. The regional cluster head further broadcasts it using the maximum transmission power to either directly deliver it to the base station or through another regional cluster head.

CHAPTER 4

IMPLEMENTATION

4.1 Simulation Scenario

The simulation was performed using the network simulator ns-2. The network field size is 1500mx1500m, containing 90 static nodes. All the nodes have the random motion with TCP traffic.

The experimentation is particularly interested in the Efficient Energy usage and service provisioning. The input to this algorithm is query with service request and destination node. The output of this algorithm is routing the query to the group member.

4.2 The Network AniMator

NAM stands for Network Animator. This tool animates the network elements. A complete visualization is available to the user which depicts the networking concepts in the project. The animator takes the tcl file as the input and creates a nam and a trace file as outputs. NAM consists of tools for editing the network topology, navigation bar and a step size controller for time. With all these tools it is possible to vividly view how actually the project works with finer resolution in time. The tools for editing include zoomer and controls for interfacing. A status bar at the bottom of the animator indicates the current status of the network elements.

In order to evaluate the proposed algorithm, two different scenarios were considered here. The area of deployed WSN is same for all scenarios and is 1500x1500m .The energy model used here is first order radio model and the In the first scenario, the base station is located at the center of the WSN. The configuration parameters for scenario1 are illustrated in Table 4.1

TABLE 4.1

CONFIGURATION PARAMETERS OF SCENARIO 1

Parameter	Value
Network size	1500×1500(m)
Base station location	(750,750)m
Number of sensor nodes	100
Initial energy	100 J
Data packet size	1024 bytes
Transmission range	250 m

In the second scenario, the base station is outside of the WSN. The configuration parameters for scenario 2 are illustrated in Table 4.2.

TABLE 4.2

CONFIGURATION PARAMETERS OF SCENARIO 2

Parameter	Value
Network size	1500×1500(m)
Base station location	(40,1200)m
Number of sensor nodes	100
Initial energy	100 J
Data packet size	1024 bytes
Transmission range	250 m

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

5.1 PERFORMANCE EVALUATION

This section presents the results of the experiments that have done to evaluate the algorithm.

5.1.1 Energy Consumption for Clustering

The cluster formation algorithm is evaluated using NS2 simulator and compared with two different clustering algorithms that are LEACH and CHEF for various WSN configurations. Experimental results have shown that EQSR algorithm performs better than LEACH, and CHEF. The energy consumption was calculated against number of rounds. In each round of the scenarios, first, clusters-heads are elected and then clusters are formed. Afterwards, each ordinary node forwards a certain bits of data to its cluster-head. Each cluster-head aggregates the received data and forwards it to the base station. The energy consumption is little bit higher while placing the base station outside of WSN.

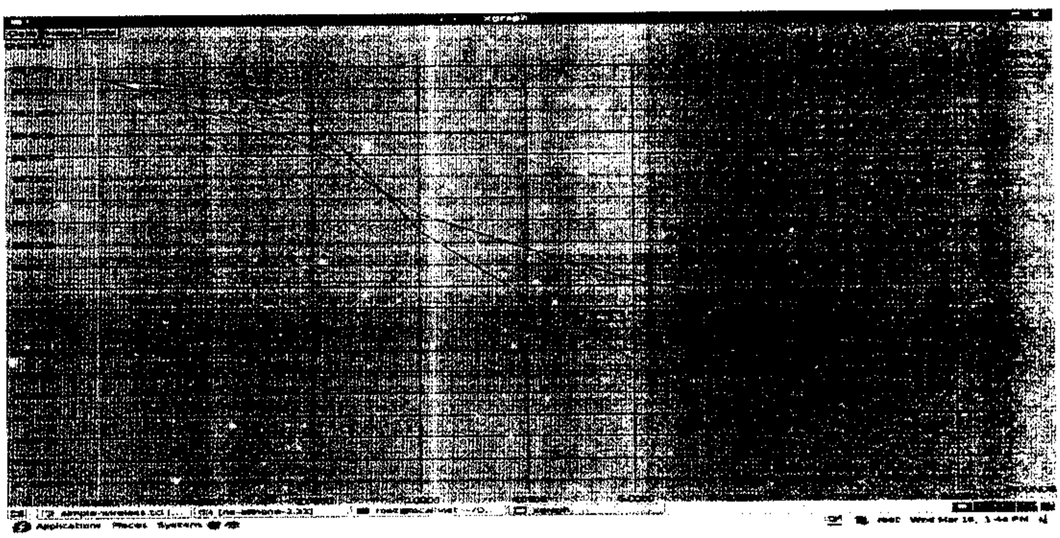


Fig 5.1: Energy consumption for clustering (scenario 1)

The experimental results are shown for both of the scenarios where the energy consumption is less for ESQR to eight rounds of cluster head election shown in the fig 5.1& 5.2.

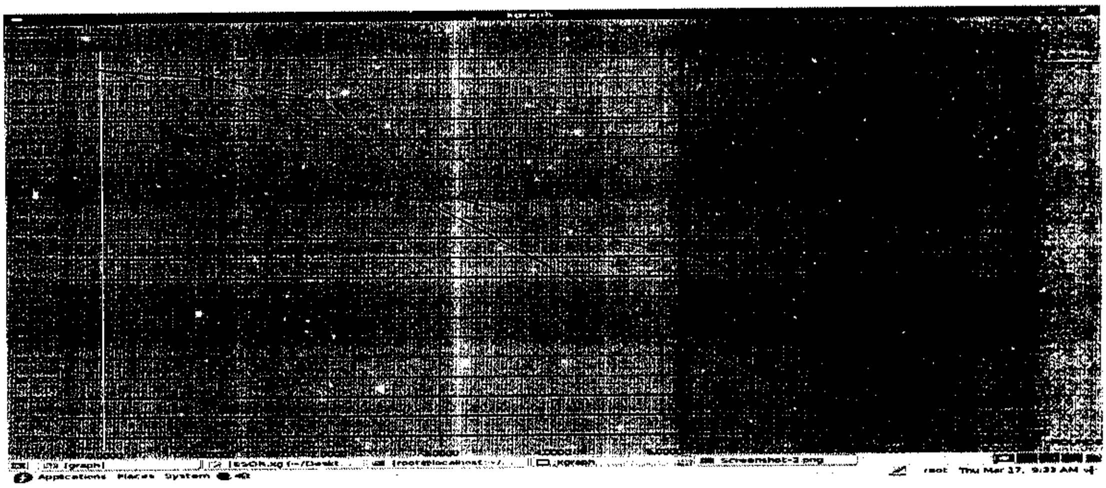


Fig 5.2: Energy consumption for clustering (scenario 2)

5.1.2 Energy Consumption for Routing

The energy consumption to route a query both in SARP and EQSR were calculated under various network sizes while maintaining the node density to 1 node/4m^2 where the nodes are evenly distributed geographically.

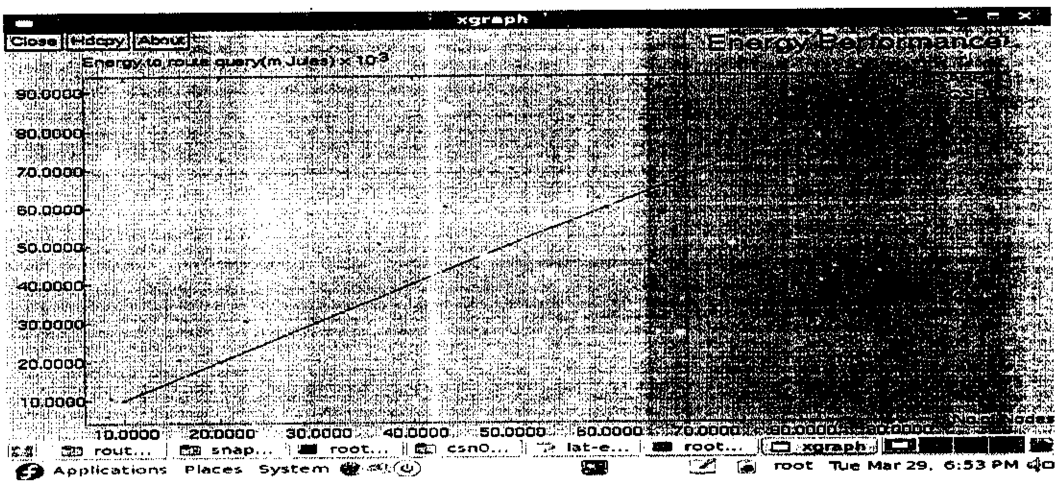


Fig 5.3: Energy consumption for routing (scenario 1)

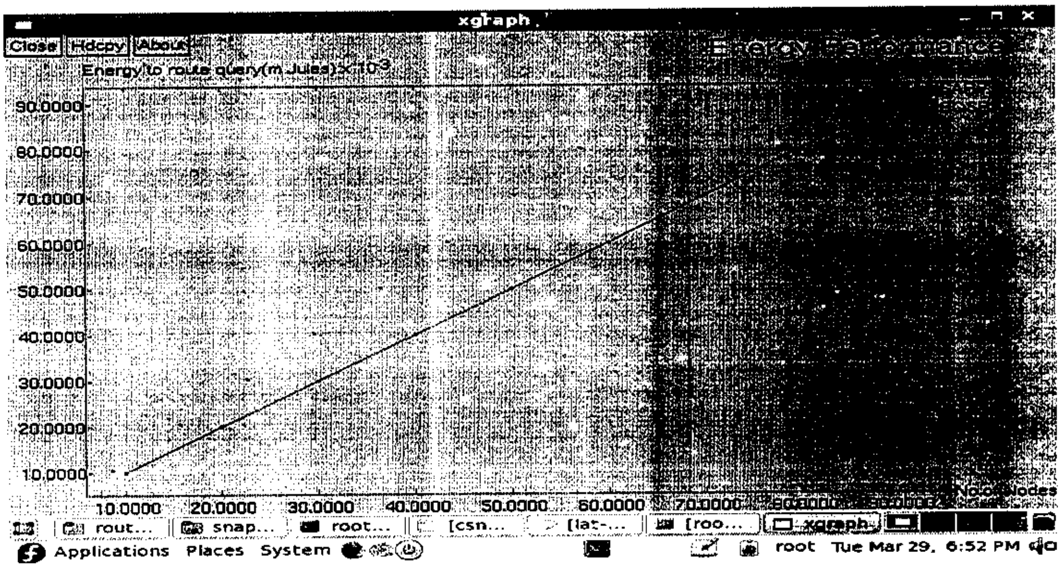


Fig 5.4: Energy consumption for clustering (scenario 2)

The result in Fig.5.3 & 5.4 reflects only the energy consumed on the path of the query and not the overall energy consumption. The Energy consumption is linearly increases for SARP. This initial evaluation result points that ESQR's overall energy consumption lower compared to SARP. And it is also more scalable as the energy consumption stays flat for a few more increases in number of nodes every now and then.

5.1.3 Transmission Latency

The latency comparison is based on the number of hops required to route a query from the base station to the farthest node when the network size increases. Number of hops is a good indicator of the latency measure used to compare various routing algorithms. The number of hops increases linearly for SARP whereas for ESQR this depends on the number of regional and service cluster heads required. In general, ESQR performs well as the queries do not have to go through each intermediate node to reach a specific node.

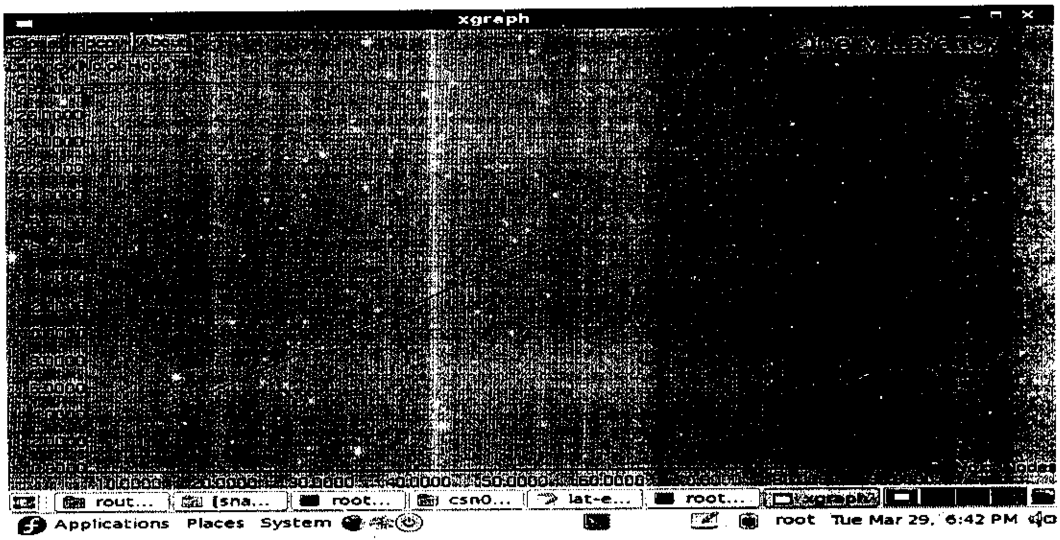


Fig 5.5: Query Latency (scenario 1)

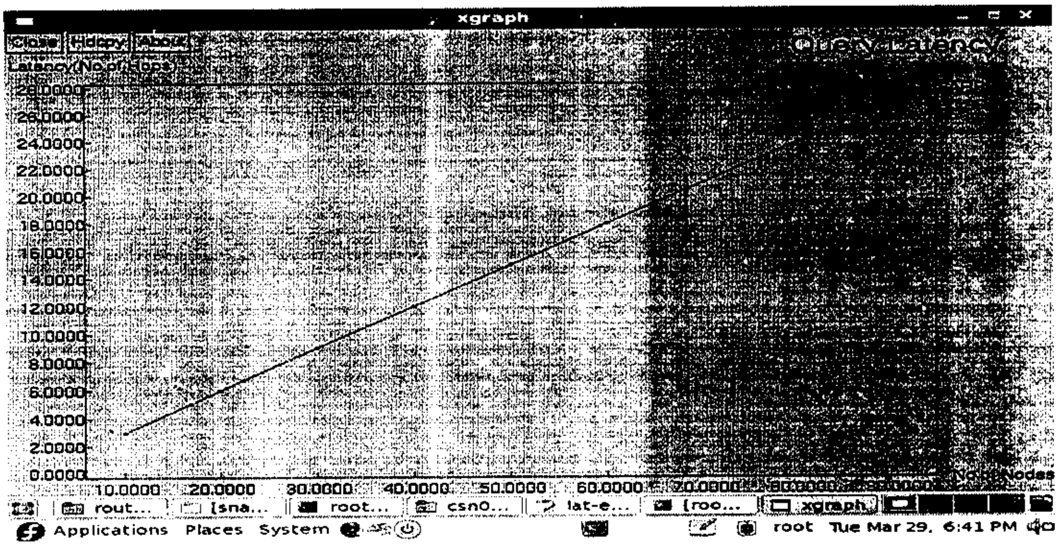


Fig 5.6: Query Latency (scenario 2)

The algorithm in ESQR uses the maximum transmission power level to reach the next regional cluster head when the query doesn't belong to the first regional cluster head that receives the query from the base station. This continues in a similar fashion until the appropriate regional cluster head is found and it would always take only two more hops to route the query from the regional cluster head to the service cluster head and then to a specific node that can provide the requested service.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This project proposes Energy efficient Service oriented query **Routing** algorithm for WSNs, namely ESQR. The main objective of our algorithm is to route the queries based on the service requested by the query with minimum energy consumption to prolong the lifetime of the WSN. To achieve this goal, this work aims to assign proper cluster-head competition ranges to sensor nodes. ESQR adjusts the cluster-head radius values considering energy and distance to the base station parameters of the sensor nodes.

ESQR routes queries to their destination based on their contents in order to allow users specify their queries without having to specify the destination of the query which is usually desirable in a WSN. This data centric routing of queries is achieved in a power efficient manner in ESQR since the network is organized as service clusters and a much longer life span for the network. The routing state in ESQR is distributed to a group of regional and service cluster heads to improve overall memory consumption and maintainability of the routing state instead of duplicating it on each and every node in the network.

The result shows that the proposed algorithm has a better performance compared to LEACH, CHEF in cluster formation and provides minimum Energy consumption with improved query latency compared to SARP in Routing while considering the simulation results. The total remaining energy level of ESQR at a certain round is higher than all the other algorithms. As a result ESQR is a stable and energy efficient routing algorithm for Service-Oriented Wireless Sensor Networks.

ESQR algorithm is designed for the WSNs that have stationary sensor nodes. As a future work this can be extended for handling mobile sensor nodes.

CHAPTER 7

APPENDICES

7.1 Source Code

```
# ==================================================================
# Define Node Configuration paramaters
#==================================================================
set val(chan)            Channel/WirelessChannel      ;#Channel Type
set val(prop)            Propagation/TwoRayGround     ;# radio-propagation model
set val(netif)           Phy/WirelessPhy              ;# network interface type
set val(mac)             Mac/802_11                   ;# MAC type
set val(ifq)             CMUPriQueue                  ;# interface queue type
set val(ll)              LL                           ;# link layer type
set val(ant)             Antenna/OmniAntenna          ;# antenna model
set val(ifqlen)          50                           ;# max packet in ifq
set val(nn)              90                           ;# number of mobilenodes
set val(rp)              TEEN                         ;# routing protocol
set val(x)               1500                         ;# X axis distance
set val(y)               1500                         ;# Y axis distance
set opt(energymodel)     EnergyModel                  ;# Initial Energy
set opt(radiomodel)      RadioModel                   ;# Transmission Model
set opt(initialenergy)   100                          ;# Initial energy in Joules

# Creating Simulator Object
set ns [new Simulator]

# Creating NAM File
set namTracefile [open clus.nam w]
$ns namtrace-all-wireless $namTracefile $val(x) $val(y)

# Creating Multiple Trace
set traceFile [open clus.tr w]
$ns trace-all $traceFile

# Creating Topology
set topo [new Topography]
set val(rp) DSR
$topo load_flatgrid $val(x) $val(y)

# Creating GOD(General Operation Director) Object
create-god $val(nn)
```

```
# Parameters
Phy/WirelessPhy set bandwidth_ 2e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Mac/802_11     set dataRate_ 2.0e6

#Creating Group-1 Nodes
for { set i 0 } { $i<=89 } { incr i } {
set node($i) [$ns node]
}

# Parameters to the Node
for { set i 0 } { $i<=89} { incr i } {
$node($i) random-motion 0
$node($i) set X_ 0.0
$node($i) set Y_ 0.0
$node($i) set Z_ 0.0
$node($i) color black
$ns initial_node_pos $node($i) 30
}

set c1 brown
set c2 tan1
set c3 purple
set c4 green4
set c5 blue
set rate 1000
set l1 T
set c6 red
set dis [open E-Distance.txt w]
puts $dis
"\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
puts $dis "\tsource-Node\tDest-Node\tE-Distance(d)\t range"
puts $dis
"\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
close $dis
set nbr [open Neighbor w]
puts $nbr
"\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
puts $nbr "\tsource-Node\tNeighbor-Node\tH-Distance(d)"
puts $nbr
"\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
close $nbr
#~~~~~~~~~~~~~~~~ For Calculation of Euclidean distance~~~~~~~~~~~~~~~~~~~~~~~
```

```
proc distance { n1 n2 nd1 nd2} {
global r
set dis [open E-Distance.txt a]
set nbr [open Neighbor a]
set x1 [expr int([$n1 set X_])]
set y1 [expr int([$n1 set Y_])]
set x2 [expr int([$n2 set X_])]
set y2 [expr int([$n2 set Y_])]
set d [expr int(sqrt(pow(($x2-$x1),2)+pow(($y2-$y1),2)))]
if {$nd2>$nd1} {
if {$d<=300} {
puts $dis "\t$nd1\t\t$nd2\t\t$d\t\tclose"
#puts $dis "close"
}
if {$d>=500} {
puts $dis "\t$nd1\t\t$nd2\t\t$d\t\tfar"
}
if {$d>300 && $d<500} {
puts $dis "\t$nd1\t\t$nd2\t\t$d\t\tmedium"
}
}
if {$d<250} {
if {$nd2!=$nd1} {
puts $nbr "\t$nd1\t\t$nd2\t\t$d"
}
}
close $dis
close $nbr
}
#~~~~~~~~~~~~~~~~~~~~~~~For Calculating the Residual Energy~~~~~~~~~~~~~~~~~~~~

proc energy {stnode etnode stime etime} {
set etp [open etmp w]
puts $etp "$stnode $etnode $stime $etime"
close $etp
exec awk -f energy.awk etmp clus.tr
exec awk -f ene.awk etmp clus.tr
exec awk -f tre.awk etmp clus.tr
exec awk -f qq.awk tre.txt
}
#~~~~~~~~~~~~~~~~~~~~For function Calling ~~~~~~~~~~~~~~~~~~~~~~~~~~~

for {set j 0} {$j<=89} {incr j} {
$ns at 8.5 "distance $node(0) $node($j) 0 $j"
}
proc attach-CBR-traffic { node sink size interval } {
```

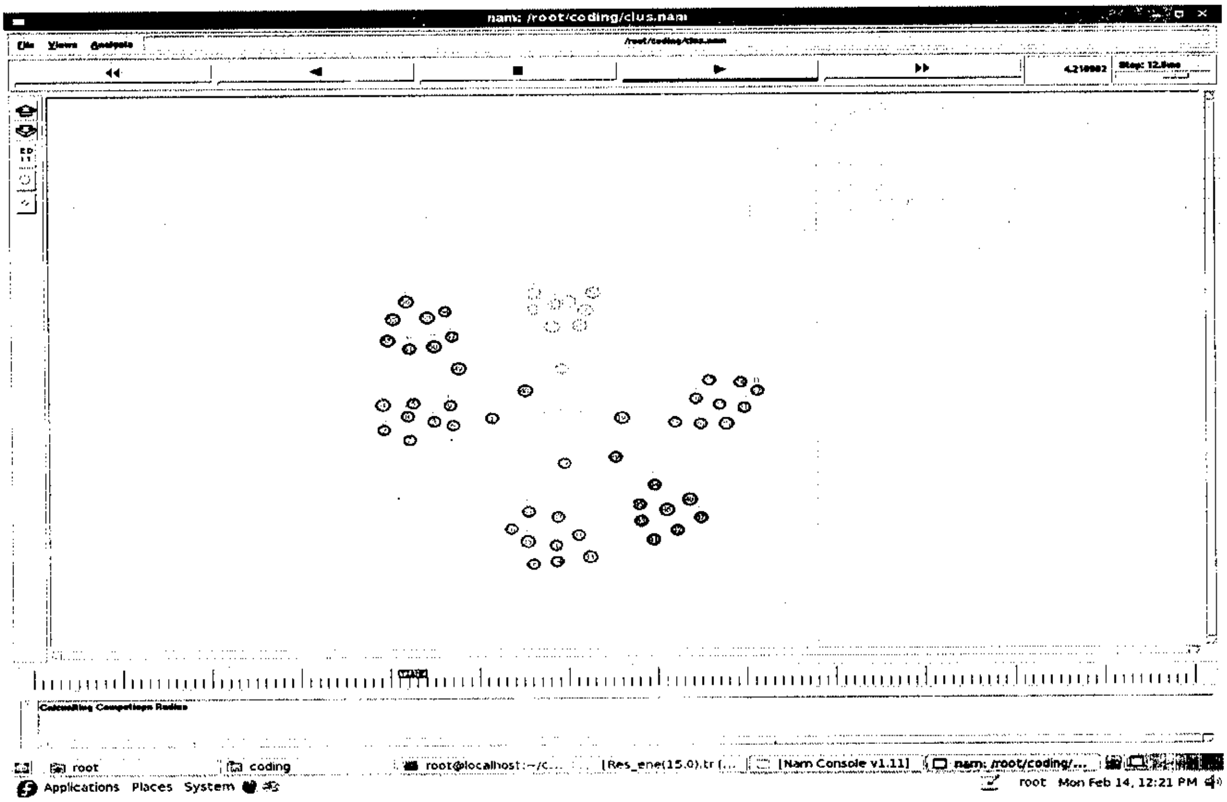
```
 #Get an instance of the simulator
 set ns [Simulator instance]
 #Create a CBR sink14 agent and attach it to the node
 set cbr [new Agent/CBR]
 $ns attach-agent $node $cbr
 $cbr set packetSize_ $size
 $cbr set interval_ $interval

 #Attach CBR source to sink;
 $ns connect $cbr $sink
 return $cbr
 }
set cb 0
set b 1

for { set i 1 } { $i<= 40} { incr i } {
set cbr($cb) [attach-CBR-traffic $node($i) $sink0 34 .042]
$ns at $k "$cbr($cb) start"
#incr i
#$ns at $k "$ns trace-annotate \"$i th node Broadcasting Residual Energy\""
$ns at $l "$cbr($cb) stop"
incr cb
}
for { set i 1 } { $i<= 89} { incr i } {
$ns at $k "$ns trace-annotate \"$i th node Broadcasting Residual Energy\""
}
#set cbr1 [attach-CBR-traffic $node(10) $sink12 32 .042]
#$ns at 8.0 "$cbr1 start"
#$ns at 8.0 "$cbr1 stop"
proc finish {} {
        global ns traceFile

        exec nam -r 2m clus.nam &

       close $traceFile
        $ns flush-trace
        exit 0
#exec xgraph energy.xg &
        }


$ns at 60.0 "finish"

puts "Start of simulation.."
$ns run
```

```
# ==================================================================
#Implementation of Routing
#==================================================================

proc attach-CBR-traffic { node sink size interval } {
  #Get an instance of the simulator
  set ns [Simulator instance]
  #Create a CBR sink14 agent and attach it to the node
  set cbr [new Agent/CBR]
  $ns attach-agent $node $cbr
  $cbr set packetSize_ $size
  $cbr set interval_ $interval

  #Attach CBR source to sink;
  $ns connect $cbr $sink
  return $cbr
 }
while {$g<=5 } {
puts "Enter the service u want
\n1.Pressure\n2.Temparature\n3.Vapour\n4.Humidity\n5.Sound\n6.Light "
gets stdin "per"
   if {$per==1} {
set cbr($cb) [attach-CBR-traffic $node(0) $sink(1) 1024 .042]
$ns at 6.02 "$cbr($cb) start"
$ns at 6.2 "$cbr($cb) stop"
incr cb
puts "Enter the node from which u want the service information \n 2 3 4 5 6 7 8 9 "
gets stdin "num"
incr cb
set k [expr $k+.1]
set l [expr $l+.1]

if {$num==2 || $num==3 || $num==4 || $num==5 || $num==6 ||$num==7 || $num==8 || $num==9
} {

#set cbr($cb) [attach-CBR-traffic $node($num) $sink1 1024 .042]
set cbr($cb) [attach-CBR-traffic $node(1) $sink($num) 1024 .042]
$ns at 6.0 "$cbr($cb) start"
set k [expr $k+1]
set l [expr $l+1]
}
incr cb
}

 if {$per==2} {
```

```
$ns at 7.4 "$cbr($cb) stop"
puts "ENter ANY ONE  \n 30.\n31. \n 32. \n33.\n34.\n35.\n36.\n37\n38.  "
gets  stdin "num"
incr cb
set k [expr $k+1]
set l [expr $l+1]
if { $num==31 || $num==32 || $num==33 || $num==34 ||$num==35 || $num==36 || $num==37||
$num==38  } {

set cbr($cb) [attach-CBR-traffic $node(29) $sink($num) 1024 .042]
$ns at 7.2 "$cbr($cb) start"
$
}
set k [expr $k+1]
set l [expr $l+1]
if { $num==40 || $num==41 || $num==42 || $num==43 ||$num==44 || $num==45 || $num==46||
$num==47  } {
set cbr($cb) [attach-CBR-traffic $node(39) $sink($num) 1024 .042]
$ns at 7.5 "$cbr($cb) start"
$ns at 7.5 "$ns trace-annotate \"$num Transmitting\""
$ns at 7.5 "$node($num) color red"
$ns at 7.7 "$cbr($cb) stop"
incr cb
set k [expr $k+1]
set l [expr $l+1]
}
}
incr g
}
set ptp [open throughput.xg w]
global  count ptp ppdr ns
set t [$ns now]
set itval 0.5
proc finish {} {
        global ns traceFile

        exec nam -r 2m clus.nam &
         exec xgraph throughput.xg &
      close $traceFile
        $ns flush-trace
        exit 0
        }

puts "Start of simulation.."

$ns run
```

7.2 Screen shots

- **Node creation**



- **Competition radius calculation**

*Txn.tr (~/Desktop/daffodils.suriya03@gmail.com/CSN0165/coding) - gedit

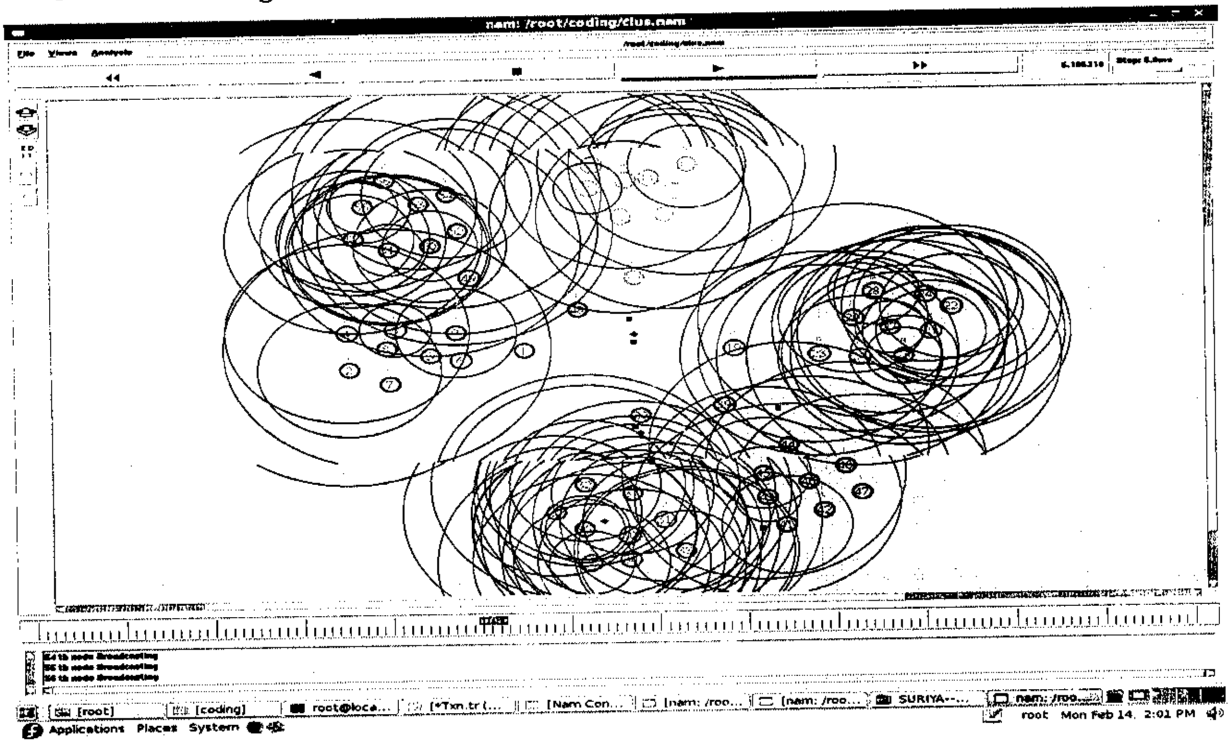
File Edit View Search Tools Documents Help

New Open Save Print... Undo Find Replace

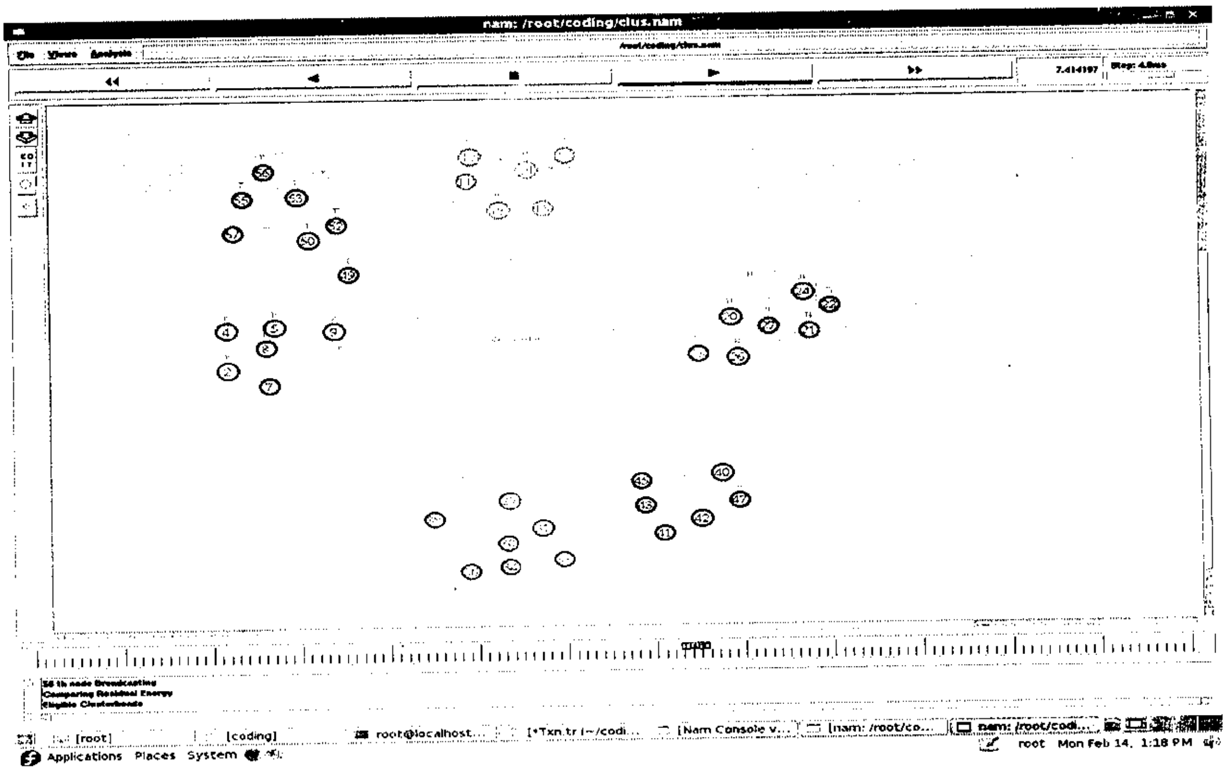
*Txn.tr

Node-Id	Distance(d)	Range	Residual-Energy	Range	Competion-Radius	TX-Range
1	112	medium	99.567921	Low	Very-small	23.45
2	128	medium	99.567951	Low	medium-small	98.22
3	220	far	99.567931	Low	Rather-Large	177.32
4	333	far	99.558789	Low	Rather-Large	167.88
5	236	far	99.567921	Low	Rather-Large	182.45
6	176	far	99.555315	Low	Rather-Large	187.43
7	249	far	99.577921	Medium	Large	201.26
8	73	close	99.579895	Medium	small	58.88
9	207	far	99.577921	Medium	Large	198.12
10	76	Close	99.567921	Low	Verysmall	24.70
11	177	far	99.587921	Medium	Large	214.52
12	187	far	99.567921	Low	Rather-Large	185.87
13	153	medium	99.577921	Medium	Medium	125.56
14	137	medium	99.567921	Low	medium-small	87.45
15	102	Close	99.577921	Medium	small	54.11
16	111	close	99.577451	Medium	small	51.68
17	119	Close	99.567921	Low	Verysmall	17.45
18	133	medium	99.567921	Low	medium-small	100.44
19	65	Close	99.577921	Medium	small	44.68
20	102	Close	99.577921	Medium	small	51.85
21	150	medium	99.567771	Low	medium-small	92.45
22	136	medium	99.567921	Low	medium-small	88.90
23	120	medium	99.568191	Low	medium-small	84.43
24	115	medium	99.587191	Medium	medium	123.76

- **Broadcasting Residual Energy**

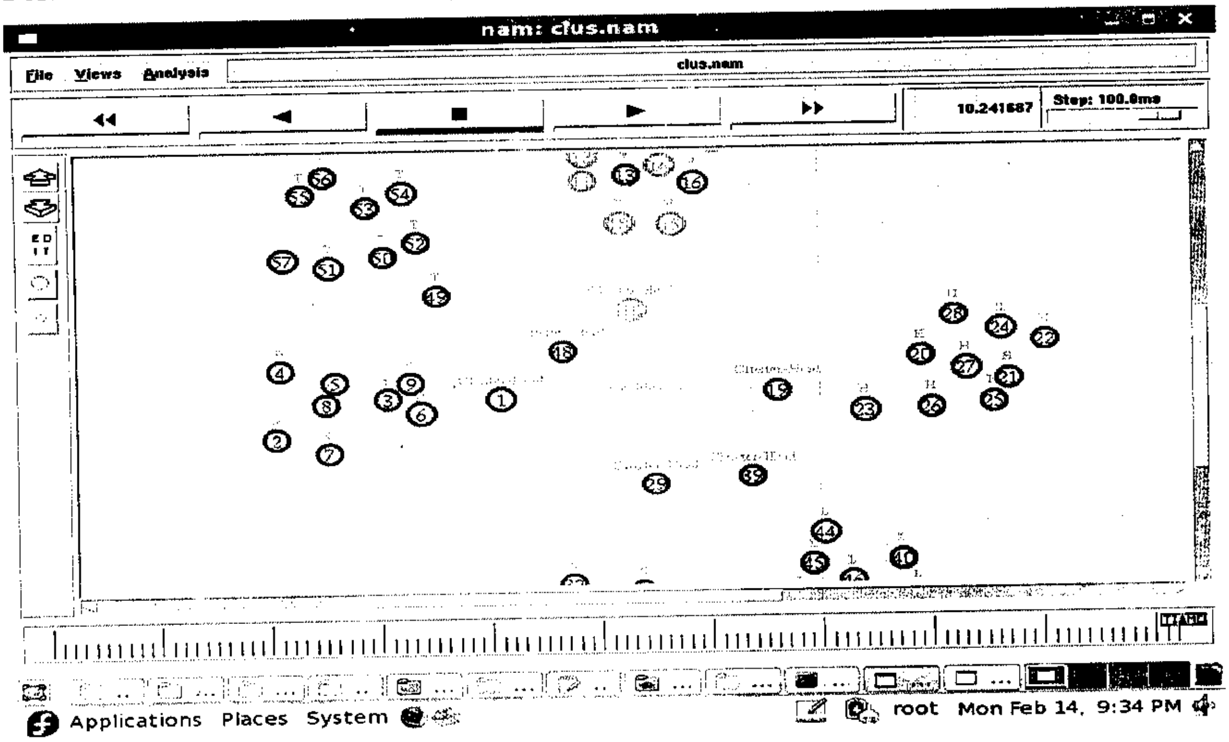


- **Eligible Cluster Heads**

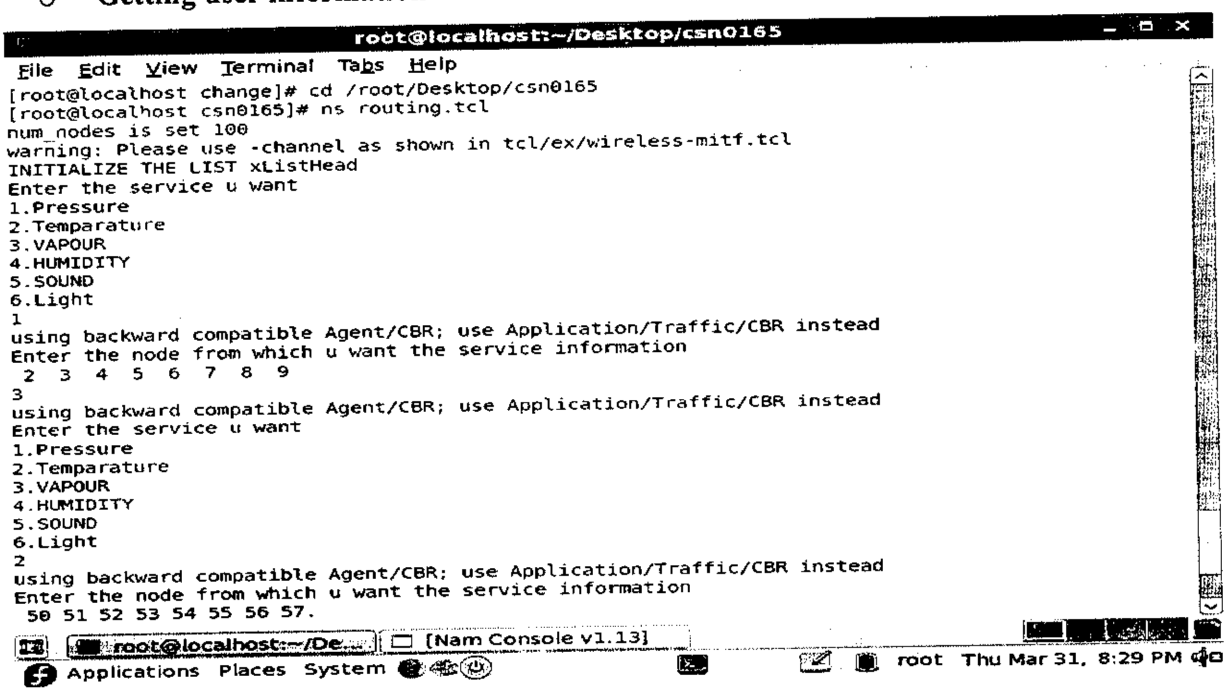


- Cluster

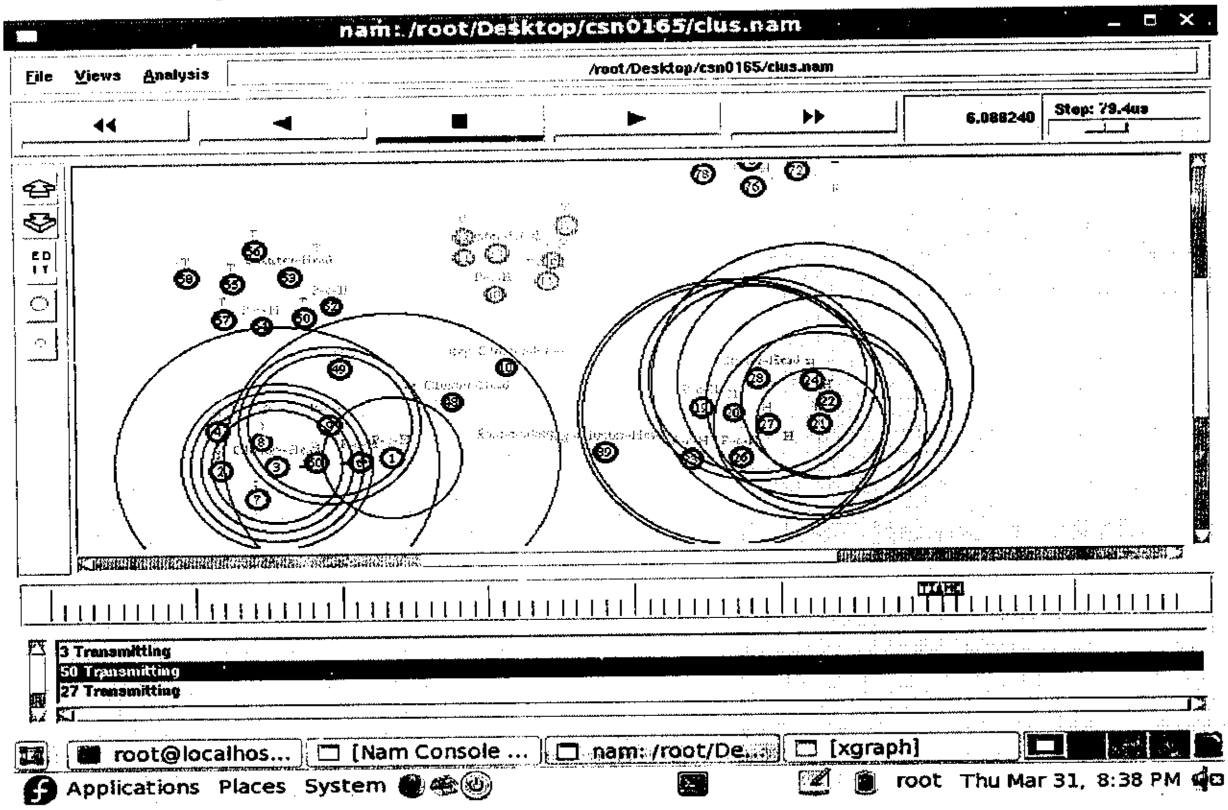
Formation



- Getting user information



- Performing Routing



REFERENCES

[1] F.L. Lewis, "Wireless Sensor Networks in Smart Environments: Technologies, Protocols, Applications" , ed. D.J. Cook and S.K. Das, Wiley, New York, 2004.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks", in the Proceeding of the Hawaii International Conference System Sciences, Hawaii,January 2000.

[3] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

[4] Hakan Bagci and Adnan Yazici, " An Energy Aware Fuzzy Unequal Clustering Algorithm for Wireless Sensor Networks" IEEE conference on Fuzzy Systems,pp.1-8,September 2010.

[5] Akyildiz. I, Pompili. D, Melodia. T, " Underwater Acoustic Sensor Networks: Research Challenges", Ad Hoc Netw. 2005, 3, 257–279.

[6] Seifu Tatek Yazhy and Seung-Kyu Park, "Hierarchical Service-Oriented Routing of Queries in Wireless Sensor Networks", Proceedings of the 2nd International Conference on Interaction Sciences, November 24-26, 2009.

[7] Kavi Kumar Khedo and R.K. Subramanian, "A Service-Oriented Component-Based Middleware Architecture for Wireless Sensor Networks", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3, March 2009.

[8] Alan J. Demers, Johannes Gehrke, Rajmohan Rajaraman, "The Cougar Project: a Work-in-Progress Report", SIGMOD Record, pp.53–59, 2003.

[9] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, "TinyDB: an acquisitional query processing system for sensor networks", ACM Trans. Database Syst, pp.122–173,2003.

[10] Daniel J. Abadi, Samuel Madden, Wolfgang Lindner, "REED: robust, efficient filtering and event detection in sensor networks" , in:Proceedings of the 31st International Conference on Very Large Data Bases, August 30–September 2 2005, Trondheim, Norway, 2005, pp.769–780.

[11] S. Dulman, T. Nieberg, J. Wu, P. Havinga, "Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks", WCNC Workshop, New Orleans, Louisiana, USA, March 2003.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffiusion: a scalable and robust communication paradigm for sensor networks", Proceedings of ACM MobiCom '00, Boston, MA, 2000, pp. 56-67.

[13] K. Sohrabi, J. Pottie, "Protocols for self-organization of a wireless sensor network", IEEE Personal Communications, Volume 7, Issue 5, pp 16-27, 2000.

[14] F.L. Lewis, "Wireless Sensor Networks in Smart Environments: Technologies, Protocols, Applications" , ed. D.J. Cook and S.K. Das, Wiley, New York, 2004.

[15] Younis O. and Fahmy S. (2004) "HEED: A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks", IEEE Transactions on Mobile Computing, vol. 3, no. 4.

[16] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks ",in IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005, p. 8.

[17] S.Lindsey,C.Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", IEEE Aerospace Conference Proceedings, 2002, Vol. 3, 9-16 pp. 1125-1130.

[18] A. Manjeshwar and D. P. Agarwal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks", In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001.

[19] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", Wireless Networks, Vol. 8, No. 5, Page(s): 481-494, September 2002.

[20] Delicato F, Pires P, Rust L, Pirmez L, de Rezende J, "Reflective middleware for wireless sensor networks". In: 20th Annual ACM symposium on applied computing (ACM SAC'2005), Santa Fe, USA, pp 730–735

[21] Rezgui A,Eltoewissy M, "Service-Driven Query routing in sensor Networks ", In 31st IEEE Conference on Local computer Networks, pp.649-655, Tampa, Florida, USA, 14-16 Nov 2006.

[22] Rezgui A, Eltoweissy M, μRACER, "A Reliable Adaptive Service-Driven Efficient Routing Protocol Suite for Sensor-Actuator Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 5, pp. 607-622, May 2009.

[23] Wendi R. H., Anantha C, and Hari B, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Proceedings of the 33rd Hawaii International Conference on Systems Sciences, Maui, Hawaii, January 2000. pp. 1-10.

[24] Ilak S, Abu-ghazaleh N and Heinzelman W, "A Taxonomy of Wireless Microsensor Network Models", in ACM Mobile Computing and Communications Review (MC2R) [C], June 2002.
[25] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, "A Two-tier data dissemination model for large-scale wireless sensor networks", proceedings of ACM/IEEE MOBICOM, 2002.