Algorithm for Energy Preservation in Wireless Sensor Network

Smit D. Saraiya
Department of Computer Science & Technology, Uka Tarsadia University Tarsadi, Mahuva-Bardoli Road, Surat, Gujarat, India

Yogendra K. Patel
Department of Computer Science & Technology, Uka Tarsadia University Tarsadi, Mahuva-Bardoli Road, Surat, Gujarat, India

Urvish D. Modi
Department of Computer Science & Technology, Uka Tarsadia University Tarsadi, Mahuva-Bardoli Road, Surat, Gujarat, India

Vatsal P. Shah
Department of Computer Science & Technology, Uka Tarsadia University Tarsadi, Mahuva-Bardoli Road, Surat, Gujarat, India

ABSTRACT

Wireless sensor network has many issues like energy consumption, security, etc. Wireless sensor network is broadly used in different field like battle-field, education and organization etc. This paper identify that, in order to further development of WSN application in real world, minimizing energy waste is one of the most life-threatening problem. Replacing this energy resource in the field is usually not realistic, and all together, a WSN must operate at least for a given mission time or as long as possible. Hence, the lifetime of a WSN becomes a very essential figure of merit. Here problem recognize by students is that in MAC protocol overhead is biggest issue for WSN. To minimize energy wastage, algorithm is designed in such manner that will decrease overhead from data transmission.

Therefore, mismanaged spares can shorten WSN lifetime instead of extending it. We measured WSN lifetime by simulating the behavior of multiple nodes.

Keywords
Wireless Sensor network, Energy Prevention, Energy management

1. INTRODUCTION
Wireless sensor networks have been the focus of considerable research during the past few years. The research issues currently addressed in wireless sensor networks are hardware constraints, communication and routing issues, data management problems, and software engineering principles. One of the most important issue apart from the above mentioned ones is energy optimization in wireless sensor networks.

2. FACTORS AFFECTING LIFETIME OF NODE
Here we discuss import factors affecting the energy of sensor node. These are basic complications; by overcoming or precaution on this area directly lead us to preventing energy.

Data Aggregation

In classic wireless sensor networks, sensor nodes are typically resource-constrained and battery-restricted. In order to prevent resources and energy, data must be aggregated to avoid overwhelming amounts of traffic in the network. The aim of data aggregation is that eliminates redundant data transmission and enhances the lifetime of energy in wireless sensor network.
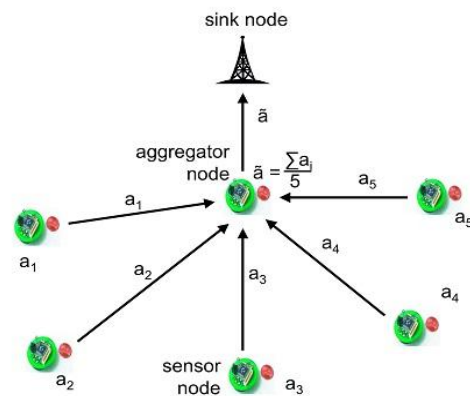

Figure 1: Data Aggregation Example

The help of data aggregation process we can enhance the robustness and accuracy of information which is obtained by entire network, certain redundancy exists in the data collected from sensor nodes thus data fusion processing is needed to reduce the redundant information.

Scheduling

A Scheduling in WSN, is called as the packet scheduling, which is used for manages the sequencing of packets in wireless sensor network of transmit and receive queues of the wireless network interface controller, which is use circular data buffer. Different schedulers are available for the every operating system.
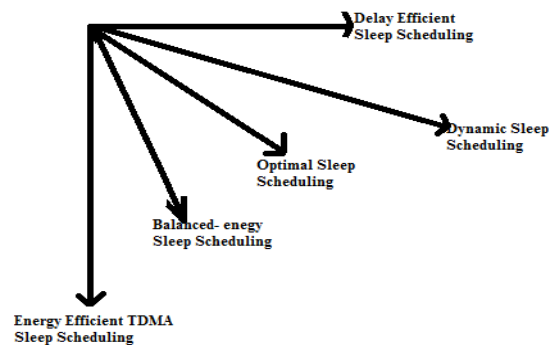

Figure 2: Scheduling Types

Wireless sensor network is used with the scheduling which definitely increases the network validity for the long period of time.

Routing Algorithms

Table 1: Routing Protocol Types and Performance in every area

Routing Protocols	Classification	Power Usage	Data Aggregation	Scala Bility	Query Based	Over head	Data delivery Model	QoS
SPIN	Flat/Src-initiated/ Data-centric	Ltd.	Yes	Ltd	Yes	Low	Event driven	No
RR	Flat	Low	Yes	Good	Yes	Low	Demand Driven	No
GBR	Flat	Low	Yes	Ltd	Yes	Low	Hybrid	No
CADR	Flat	Ltd		Ltd	Yes	Low	Continuously	No
COUGAR	Flat	Ltd	Yes	Ltd	Yes	High	Query driven	No
LEACH	Hierarchical/ Dst-initiated /Node-centric	High	Yes	Good	No	High	Cluster-head	No
TEEN & APTEEN	Hierarchical	High	Yes	Good	No	High	Active Threshold	No
SOP	Hierarchical	Low	No	Good	No	High	Continuously	No
GAF	Hierarchical/ Location	Ltd	No	Good	No	Mod	Virtual grid	No
SPAN	Hierarchical/ Location	Ltd	Yes	Ltd	No	High	Continuously	No
GEAR	Location	Ltd	No	Ltd	No	Mod	Demand Driven	No
SPEED	Location/Data Centric	Low	No	Ltd	Yes	Less	Geographic	Yes

MAC protocol research

> - **Over hearing**
> - **Overhead**
> - **Idle listening**
> - **Collision**

The **hidden terminal problem** is due to the fact that a node (say A) transmitting to another node (say B) cannot hear transmissions from another node C, which might also be transmitting to B, and might interfere with the A-to-B transmissions. E.g. Say transmissions are limited to a distance d; this arrangement causes the problem described:
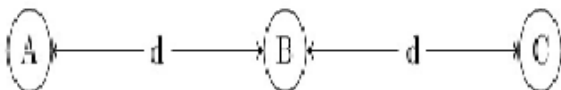

Figure 3: Hidden terminal problem [1]

Control Packet Overhead: For hidden terminal problem controlling packets (RTS/CTS) mechanism introduce. But somehow sending (RTS) and receiving (CTS) control packets consumes energy too, and less useful data packets can be transmitted. Minimal number of control packets should be used to achieve data transmission [19].

Algorithm

string d=""

Collision: When a transmitted packet is corrupted it has to be discarded, and the follow on retransmissions increase energy consumption. Collision increases latency also.

Idle Listening: Listening to receive possible traffic that is not sent can consume extra energy. If nothing is sensed, the sensor node will be in idle state for most of the time.

Overhearing: Overhearing means that a node picks up packets that are destined to other nodes scan unnecessarily consume energy. Reason of energy waste is overhearing, a node receives packets that are destined to other nodes [18].

These are some of problems listed above. There may be possibility of other problems that we don't know. Some of problems like Collision, idle listening and overhearing that are easily reduce by scheduling, active-sleep state and RTS-CTS control packets. So our major concern is to overcome overhead problem.

3. PROPOSED SOLUTION

In this work, we describe a simple algorithm for energy management in WSN. In above section we describe energy management methods and MAC layer issues and try to overcome these issues so that node and network life can increase. We are in a channel to develop algorithm for energy management for WSN to solve the problem of energy.

tc=5

while(state) do

    if(state=active) then

        tc=tc-1

        // sense data

        d=d+new data

        if(sizeOf(d)>16 bytes or tc=0) then

            //control packets (RTS - CTS)

            send data

            if(ack) then

                d=""

                tc=5

            end if

        end if

    end if

    else

        node in sleep mode

end while

Description:

At startup phase buffer d initialize with null and time counter tc as (user desire value). Whenever node comes in active state timer counter tc will automatically decremented by 1. Node sense data are transmitted to variable d. Before sending data to other nodes, nodes will check condition ((d)>16 bytes or tc=0) that must satisfy. This condition determines data d must have size larger than 16bytes (user desire size) or time counter

tc value is zero. Otherwise transmission is cancelled. Before sending data control packets (RTS - CTS) need to communicate. Whenever acknowledgment is received successfully the buffer d will initialize as null and timer counter tc will initialize as 5 (user desire value).

Advantage:

Improve Energy Efficiency. Sending of control packets are restricted so that multiple tasks are reduced and energy is saved with variable.

Limitation:

It will not work for real time application. (Time critical application) Which application is data centric or mission critical applications may use this.

4. 4. RESULT AND ANALYSIS
Execution and Implementation of WSN

In this chapter we try to implement our algorithm.

Simulate network program ns2 software used. For c++ program Dev-C++ software is use.

Overhead problem simulate in VMware workstation. Here graph generates that shows overhead [26].

Problems:

1 Implement algorithm in ns2: No buffer or gathered data calculation functionality

2 Implement algorithm in c++ and bind c++ to ns2: Buffer/ Data size calculated in c++ but ns2 and c++ integration wasn't successful

RTS Packet | **CTS Packet**
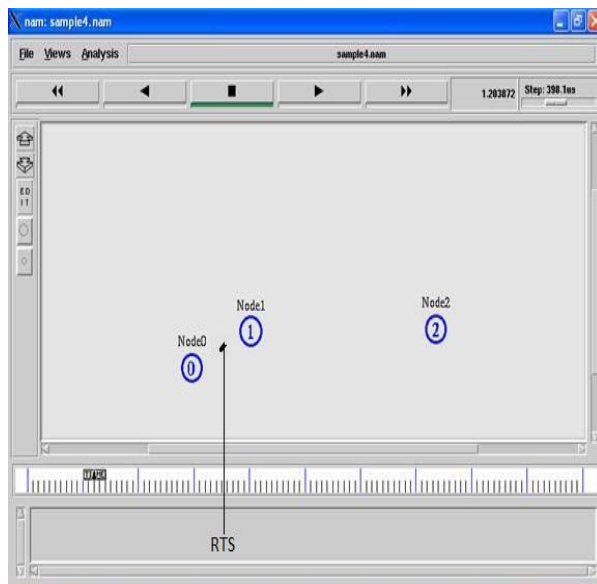

Figure 4.1: Send RTS [26] | **Figure 4.2: Send CTS [26]**

Here, in this figure 4.1 describe node1 send RTS to node0 before doing actual data transmission.

Here, in this figure 4.2 describe node0 send CTS to node1 before doing actual data transmission.

Data Send Routing overhead graph:
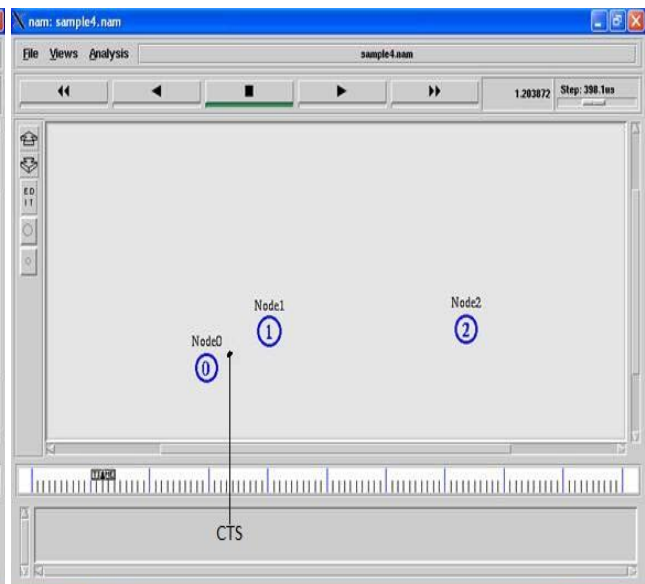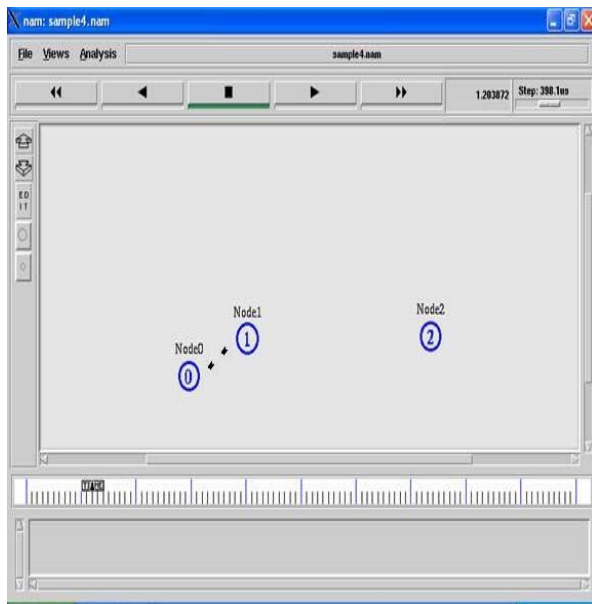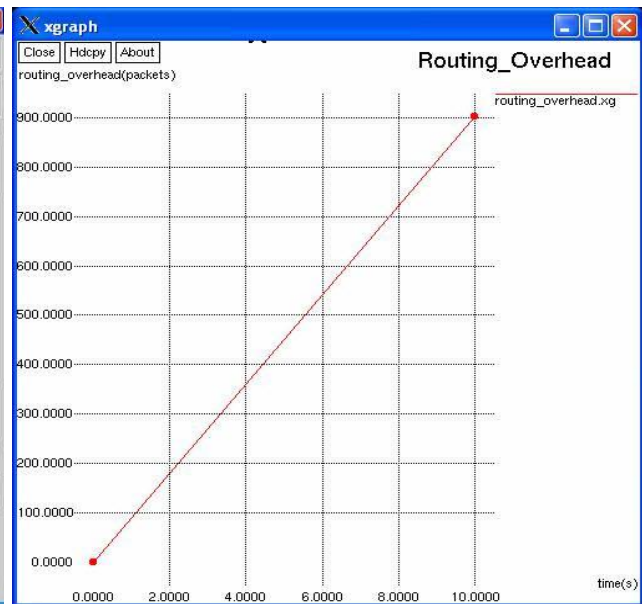
Figure 4.3: Data send [26]

Figure 4.4: Overhead graph [26]

Here, in this figure 4.3 describe node1 send DATA to node0.after control packet communication.

At initial phase no data overhead happen, but after time period control packets continuous transmitted that shows in figure 4.4.
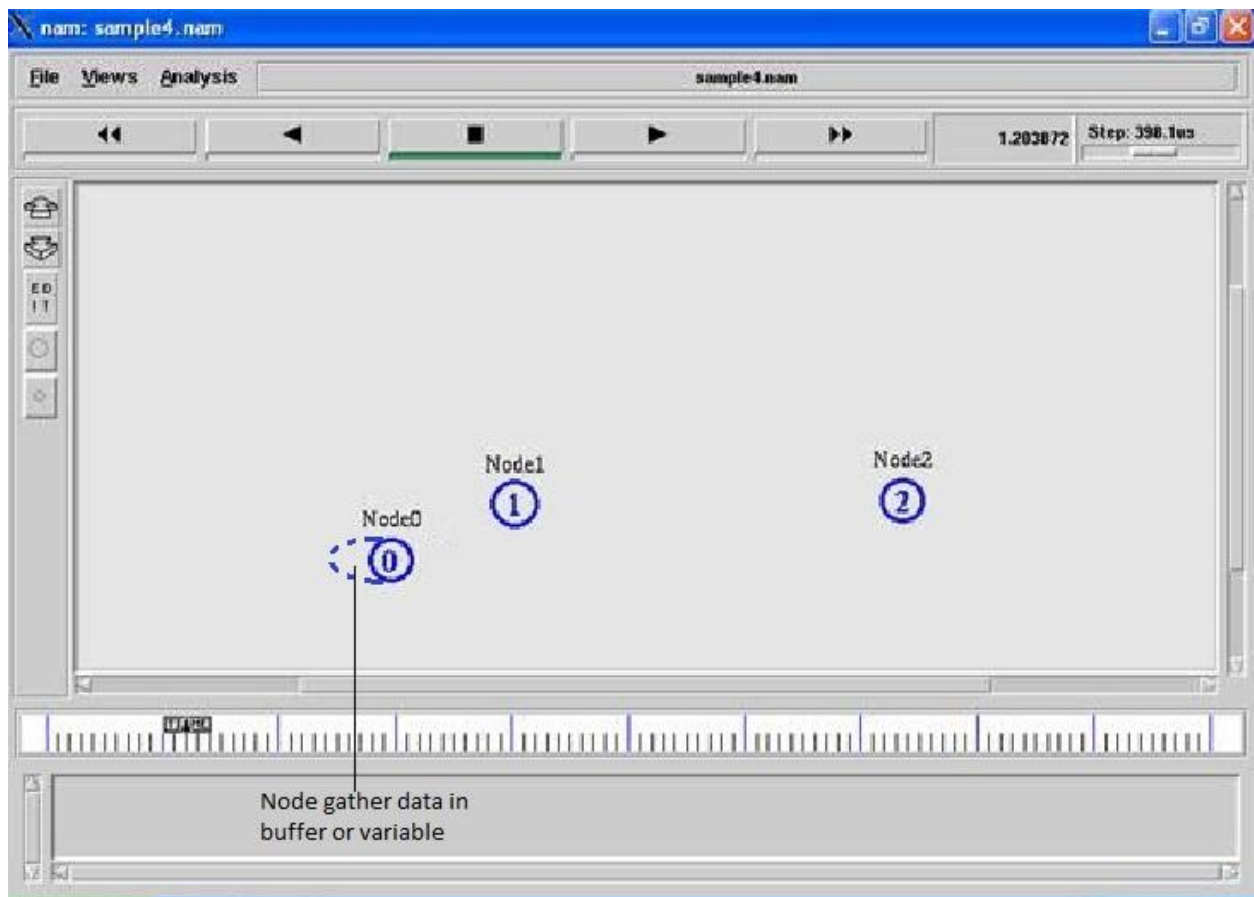
Desired situation should be:

Figure 4.5: Node store data in buffer

Node can't send sensed data directly to another node. Instead of sending data, node will wait till condition achieved. After condition becomes true data will send.

Node collects its own data till condition true, whenever condition satisfies data automatically sent.

Ns2 codes and c++ codes need to be put on paper

Future scope and next work can possible

Program 1 C++ code shows that data won't pass if condition does not satisfy:

```cpp
#include<iostream>
#include<string>

#define MIN_SIZE 16
using namespace std;

int main()
{
string str;
int tc=5;
string
stat[]={"a","s","a","s","s","a","s","s","s","a","s","a","s","a","s"
,"s","s","a","s","s","a","s","a","s","s","s","a","s","s","a"};

string buff;
for(int i=0;i<30;i++)
{
        if(stat[i]=="a")
        {
                tc=tc-1;
                cout << "Enter Data:";
                cin >> str;
                buff.append(str);
                if(buff.length()>MIN_SIZE || tc==0)
                {
                        cout << buff<<"\n";
                        buff="";
                        cout << "data is send"<<"\n\n";
                }
        }
        else
        {
                cout<<"node in sleep mode"<<"\n";
        }
}
return 0;
}
```

This code simulates size of buffer/ string can only be sent if size exceeds condition. Here we take conditions on our assumption. User desire condition can take. This simulation shows concept like buffer.

Program 2: NS2 code performing wastage of energy and data [25]

```tcl
# Simulator Instance Creation
set ns [new Simulator]
#Fixing the co-ordinate of simulation area
set val(x) 500
set val(y) 500
# Define options
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)        Queue/DropTail/PriQueue ;# 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)         3                       ;# number of mobilenodes
set val(rp)         AODV                    ;# routing protocol
set val(x)          500                     ;# X dimension of topography
set val(y)          500                     ;# Y dimension of topography
set val(stop)                   10.0        ;# time of simulation end
# set up topography object
set topo    [new Topography]
$topo load_flatgrid $val(x) $val(y)
#Nam File Creation  nam - network animator
set namfile [open sample29.nam w]
#Tracing all the events and cofiguration
$ns namtrace-all-wireless $namfile $val(x) $val(y)
#Trace File creation
set tracefile [open sample29.tr w]
#Tracing all the events and cofiguration
$ns trace-all $tracefile
# general operational descriptor- storing the hop details in the network
create-god $val(nn)
# configure the nodes
    $ns node-config -adhocRouting $val(rp) \
                        -llType $val(ll) \
                        -macType $val(mac) \
                        -ifqType $val(ifq) \
                        -ifqLen $val(ifqlen) \
                        -antType $val(ant) \
                        -propType $val(prop) \
                        -phyType $val(netif) \
                        -channelType $val(chan) \
                        -topoInstance $topo \
                        -agentTrace ON \
                        -routerTrace ON \
```

```
                    -macTrace OFF \
                    -movementTrace ON
```

Node Creation

```
for {set i 0} {$i < 3} {incr i} {
        set node_($i) [$ns node]
        $node_($i) color black
}
```

#Location fixing for a single node

```
        $node_(0) set X_ 258
        $node_(0) set Y_ 222
        $node_(0) set Z_ 0
for {set i 1} {$i < 3} {incr i} {
        $node_($i) set X_ [expr rand()*$val(x)]
        $node_($i) set Y_ [expr rand()*$val(y)]
        $node_($i) set Z_ 0
}
```

Label and coloring

```
for {set i 0} {$i < 3} {incr i} {
        $ns at 0.1 "$node_($i) color blue"
        $ns at 0.1 "$node_($i) label Node$i"
}
```

#Size of the node

```
for {set i 0} {$i < 3} {incr i} {
        $ns initial_node_pos $node_($i) 30
}
```

```
#*****************************Defining
Communication Between node0 and all nodes
*****************************8
```

```
for {set i 1} {$i < 3} {incr i} {
```

Defining a transport agent for sending

`set udp [new Agent/UDP]`

Attaching transport agent to sender node

`$ns attach-agent $node_($i) $udp`

Defining a transport agent for receiving

`set null [new Agent/Null]`

Attaching transport agent to receiver node

`$ns attach-agent $node_(0) $null`

#Connecting sending and receiving transport agents

`$ns connect $udp $null`

#Defining Application instance

`set cbr [new Application/Traffic/CBR]`

Attaching transport agent to application agent

`$cbr attach-agent $udp`

#Packet size in bytes and interval in seconds definition

`$cbr set packetSize_ 512`

`$cbr set interval_ 0.1`

data packet generation starting time

`$ns at 1.0 "$cbr start"`

data packet generation ending time

`#$ns at 6.0 "$cbr stop"`

`}`

ending nam and the simulation

`$ns at $val(stop) "$ns nam-end-wireless $val(stop)"`

`$ns at $val(stop) "stop"`

#Stopping the scheduler

`$ns at 10.01 "puts \"end simulation\" ; $ns halt"`

`#$ns at 10.01 "$ns halt"`

```
exec awk -f Routing_Overhead.awk sample29.tr >
routing_overhead.tr &
```

#Starting scheduler

`$ns run`

This code does not simulate energy preservation, but by this way we can simulate overhead problem.

5. CONCLUSION

This dissertation work involves developing and implementation of energy reduction approach that try to overcome/ reduce overhead problem of MAC protocol. That will directly increase lifetime of mote. This approach is developed on basis of buffer or data size calculation. That limits overhead problem. Unfortunately, we do not simulate resultant scenario. Reason behind not simulating is ns2 not providing size calculation or data management mechanism.

6. FUTURE SCOPE

We have only give a focus on MAC protocol issue for energy, there is possibility of other aspect can affect energy of node. Our algorithm is focusing on size of data need to transmit, that may chance of similar approach that can save energy wastage.

This overhead problem may easily solve in future if size calculation does or c++ script can integrated with ns2.

If the perfect implementation of this approach can does, that directly leads to betterment of country as well as for all individuals. And lots of human power can redirect to other place using WSN applications.

7. REFERENCES

[1] Dr. Sunil Kumar S. Manvi, M. S. Kakasageri - Wireless & Mobile Networks-Wiley Publications

[2] Fei HU and Xiaojun Cao - Wireless Sensor Networks - CRC Press.

[3] Soumitra D., Pramod D G.," Energy Efficient Cluster based Hierarchical Routing Protocols in Wireless Sensor Network – a Survey ", Multidisciplinary Journal of Research in Engineering and Technology –vol. 1,06-22, April 2014.

[4] Urmila A. Patil, Smita V. Modi, Suma B.J., " Analysis and Implementation of IEEE 802.11 MAC Protocol for Wireless Sensor Networks", International Journal of Engineering Science and Innovative Technology (IJESIT) Vol. 2, Issue 5, September 2013.

[5] Monica R., Savan K., "A Study on Energy Efficient Routing Protocols In Wireless Sensor Networks", International Journal of Distributed and Parallel Systems Vol.3, No.3, May 2012.

[6] Amit Sharma, Kshitij Shinghal, Neelam Srivastava, Raghuvir Singh, "Energy Management for Wireless Sensor Network Nodes", International Journal of

Advances in Engineering & Technology, Vol. 1, 7-13, Mar 2011.

[7] Gilbert P., Kaliaperumal B. and Elijah R., "Research Issues In Wireless Sensor Network Applications: A Survey", International Journal of Information and Electronics Engineering-Vol. 2, No. 5, 702-706, September 2012.

[8] Kumari B., Shukla J., "Secure Routing in Wireless Sensor Network", International Journal of Advanced Research in Computer Science and Software Engineering-Volume 3, Issue 8, 746-751, August 2013.

[9] Alsoufi D., Elleithy K., Abuzaghleh T. and Nassar A., "Security in Wireless Sensor Networks – Improving The Leap Protocol", International Journal of Computer Science & Engineering Survey (IJCSES) - Vol.3, No.3, 001-013, June 2012.

[10] Kaler B. and Kaler M., "Challenges in Wireless Sensor Networks", World Wide Web (WWW).

[11] Ugur C., Andrew F. and Ye S., "Power-Efficient Data Dissemination in Wireless Sensor Networks", World Wide Web (WWW).

[12] Al-Sakib K. P, Hyung L., Choong H., "Security in Wireless Sensor Networks: Issues and Challenges", WWW

[13] Simarpreet K. and Leena M., "Power Saving Mac Protocols for Wsns And Optimization of S_Mac Protocol", International Journal of Radio Frequency Identification and Wireless Sensor Networks, Vol. 1,Page 1-8, Jun-July 2011.

[14] Wei Y., John H., Deborah E., "An Energy-Efficient Mac Protocol for Wireless Sensor Networks", WWW

[15] Subodh K., Nitin G., "Energy Efficient Deployment Techniques for Wireless Sensor Networks", International

Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 7, Page 257-262, July 2012.

[16] Shweta A., Varsha J. , Kuldeep G., "Energy Efficient Mac Protocols for Wireless Sensor Network", International Journal on Computational Sciences & Applications, Vol.4, No.1, Page 153-157, February 2014.

[17] Ilker D., Cem E. and Fatih A., "Mac Protocols for Wireless Sensor Networks: A Survey", WWW

[18] Lamia C. and Lotfi K., "Wireless Sensors Networks Mac Protocols Analysis", Journal of Telecommunications, VOLUME 2, ISSUE 1, Page 42-48, April 2010.

[19] Rajesh C., Dr. Sonia V., "Review Paper on Energy-Efficient Protocols In Wireless Sensor Networks", IOSR Journal of Engineering, Vol. 4, Issue 2, Page 01-07, Feb 2014.

[20] Rajesh Y., Shirshu V., Malaviya N., "A Survey of Mac Protocols for Wireless Sensor Networks", UbiCC Journal, Volume 4, Number 3, Page 827-833, August 2009.

[21] http://en.wikipedia.org/wiki/Wireless_sensor_network

[22] http://en.wikipedia.org/wiki/Sensor_node

[23] http://iseeindia.com/2011/12/22/wireless-sensor-networks-how-indian-universities-are-exploiting-a-new-technology/

[24] http://searchdatacenter.techtarget.com/

[25] http://slogix.in/final-year-NS2-projects/index.html

[26] https://www.amrita.edu/research/project/winsoc-%E2%80%93-wireless-sensor-networks-self-organization-capabilities-critical-and