Routing & Performance Optimization Using LEACH Protocol in WSN

Nitin D.Alety, Kiran H. Hole, Dinesh B. Parmar, K.R.Akhil, Prof. Soumitra S.Das

Abstract*—Wireless sensor networks have emerged in the past decade as a result of recent advances in microelectronic system fabrication, wireless communications, integrated circuit technologies, microprocessor hardware and nano-technology, progress in ad- hoc networking routing protocols, distributed signal processing, pervasive computing and embedded systems.. This project is an effort in designing of energy efficient Wireless Sensor Network (WSN) routing protocol. Efficient routing in a sensor network requires that the routing protocol must minimize energy dissipation and maximize network life time. Report discusses LEACH protocol. Therefore, optimal consumption of energy for WSN protocols seems essential.The purpose of this paper is to create a simulation of LEACH protocol using NS2. Here the purpose is to generate some dynamic nodes with random coordinates, and then forming cluster among them using Distance formula. Followed by random cluster head formation transmitting and receiving data to show how LEACH works.

***Keywords* - Wireless Sensor Network, Hierarchical Routing, Clustering, Energy Efficiency, LEACH protocol.**

1. INTRODUCTION

Development of energy efficient Wireless Sensor Network (WSN) Routing protocols is nowadays main area of interest amongst researchers. This project is an effort in designing of energy efficient Wireless Sensor Network (WSN) routing protocols, under certain parameters consideration. Report discusses various existing WSN routing protocols.Therefore, optimal consumption of energy for WSN protocols seems essential.

Sensor networks have emerged as a promising tool for monitoring the physical worlds, utilizing self-organizing networks of battery-powered wireless sensors that can sense, process and communicate. Wireless sensor networks consist of small low power nodes with sensing, computational and wireless communications capabilities that can be deployed randomly or deterministically in an area from which the users wish to collect data. Typically, wireless sensor networks contain hundreds or thousands of these sensor nodes that are generally identical. These sensor nodes have the ability to communicate either among each other or directly to a base station (BS). The sensor network is highly distributed and the nodes are lightweight. Intuitively, a greater number of sensors will enable sensing over a larger area [3].

2. RELATED WORKS

There are many techniques proposed as new modifications for LEACH to provide more security and to reduce energy consumption. In new approach while sequentially assigning the cluster head, the power consumption for each node; when it will be cluster head is calculated and then some results have been taken out from some simple calculations over the individual power consumption. Finally, with the help of results, the number of cluster head chance for a node will be in reverse order of Power consumption, i.e. the node at larger distance (consumes more power) will be cluster head for lesser number of times rather than a node at a shorter distance (consumes lesser power)[1].

3. GOAL

The objective is to create a simulation of LEACH protocol using NS2 & creating clusters dynamically. The aim is transmission of data with less loss of energy and increasing the lifetime of network. We are using DSDV routing protocol, which is an inbuilt protocol in NS2 which finds the shortest path for sending data between nodes.

The main aim is transmission of data in an energy efficient manner.

4. LEACH Protocol

Low-Energy Adaptive Clustering Hierarchy (LEACH) [1] is a clustering based protocol to collect data from wireless network. In the network, hundreds and thousands of wireless sensors are dispersed that collects and transmit data. Also in these networks cluster heads are elected out of the sensors to transmit the data collected. each of the sensor nodes being inexpensive and simple, their power level is low cannot be replaced and because of this, each sensor must take its turn as being a cluster head to make the protocol energy efficient.

After researching on LEACH, it was time to create a program to simulate the actual protocol. Just to get a feel of how LEACH operates the network created was very simple. Although the amount of sensor nodes and percentage of cluster heads were in small numbers, the total number of rounds varied.

The program created started with just generating nodes for the network to simulating data transmitting and receiving, but it does not have any specific transmission range.

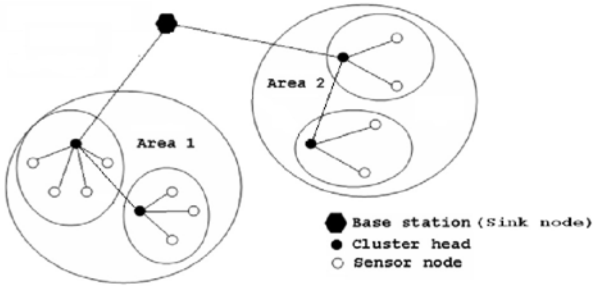


Fig.4.1: Clustering Nodes

The advantages of LEACH include the following [5]:

Any node that served as a CH in certain round cannot be selected as the CH again, so each node can equally share the load imposed upon CHs to some extent.

Utilizing a TDMA schedule prevents CHs from unnecessary collisions.

Cluster members can open or close communication interfaces incompliance with their allocated time slots to avoid excessive energy dissipation.

There exist a few disadvantages in LEACH as follows [4]:

It performs the single-hop inter-cluster, directly from CHs to the BS, routing method, which is not applicable to large-region networks. It is not always a realistic assumption for single-hop inter-cluster routing with long communication range. Besides, long-range communications directly from CHs to the BS can breed too much energy consumption.

Despite the fact that CHs rotation is performed at each round to achieve load balancing, LEACH cannot ensure real load balancing in the case of sensor nodes with different amounts of initial energy, because CHs are elected in terms of probabilities without energy considerations. Sensor nodes, with lower initial energy, that act as CHs for the same number of rounds as other sensor nodes, with higher initial energy, will die prematurely. This could bring about energy holes and coverage problems.

The idea of dynamic clustering brings extra overhead. For instance, CH changes and advertisements may diminish the gain in energy consumption.

5. TECHNICAL SPECIFICATIONS

5.1 NS2

NS (from network simulator) is a name for series of discrete event network simulators, specially ns-1, ns-2 & ns3.Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols.

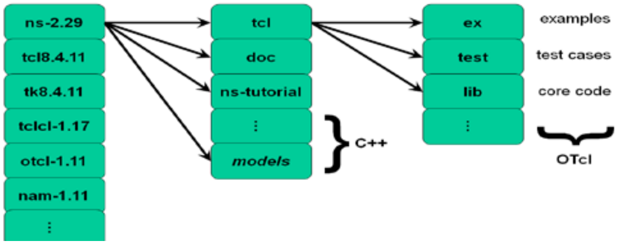


Fig 5.1: NS2 Components

NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkley written in C++ and OTcl. NS2 Creates simulation and Describe network, protocols, sources, sinks and it also Interface via OTcl which controls C++.

NS2 Execute simulation, Simulator maintains event list (packet list), executes next event (packet), repeats until done, here Events happen instantly in virtual time but could take arbitrarily long real time e.g. Single thread of control, no locking, races, etc Post-process results of NS2 are Scripts (Awk, Perl, Python) to process text output and No standard library but some available on web.

Languages in NS2:

C++ for data (creation of objects):

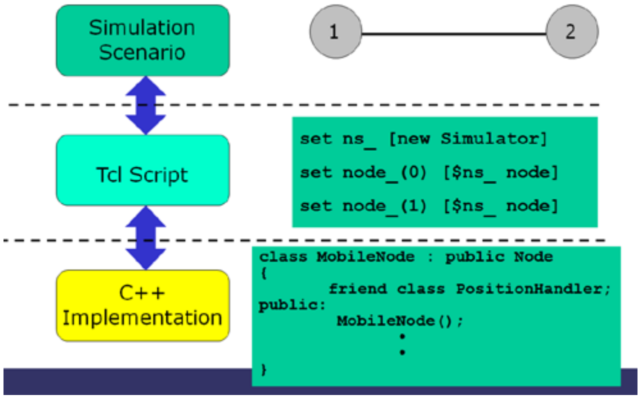
- Per-packet processing, the core of ns.
- Fast to run, detailed, complete control.
- In C++ Detailed protocol simulations require systems programming language.
- Byte manipulation, packet processing, algorithm implementation.
- Run time speed is important.
- Turnaround time (run simulation, find bug, fix bug, recompile, re-run) is slower.

OTcl for control (setup the simulator):

In Tcl Simulation of slightly varying parameters or configurations

- Simulation description
- Periodic or triggered actions
- Manipulating existing C++ objects
- Faster to write and change.
- quickly exploring a number of scenarios
- iteration time (change the model and re-run) is more important

- Turnaround time (run simulation, find bug, fix bug, recompile, re-run) is slower.



```
set ns_ [new Simulator]
set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
```

```
class MobileNode : public Node
{
        friend class PositionHandler;
public:
          MobileNode();
                    :
                    :
}
```

Fig.5.2: NS2 environment

5.2 NAM

Nam stands for Network Animator. Nam is a Tcl/Tk based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools. The network animator \nam" began in 1990 as a simple tool for animating packet trace data. This trace data is typically derived as output from a network simulator like ns or from real network measurements e.g. using tcpdump.

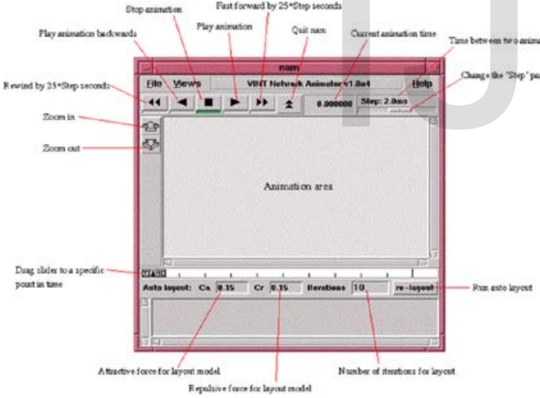

Stop animation
Fast forward by 25*Step seconds
Play animation backwards
Play animation
Quit nam
Current animation time
Time between two animation "frames"
Rewind by 25*Step seconds
Change the "Step" parameter
nam
File
Views
VINT Network Animator v1.0a4
Help
0.000000
Step: 2.0ms
Zoom in
Zoom out
Animation area
Drag slider to a specific point in time
Auto layout:
Ca
0.15
Cr
0.15
Iterations
10
re-layout
Run auto layout
Attractive force for layout model
Repulsive force for layout model
Number of iterations for layout

Fig.5.3: NAM basic view

6. IMPLEMENTATION AND SIMULATION

6.1 Cluster Formation

Cluster is used broadly in computer networking to refer to a number of different implementations of shared computing resources. Typically, a cluster integrates the resources of two or more computing devices (that could otherwise function separately) together for some common purpose.

Sensor nodes typically use irreplaceable power with the limited capacity, the node's capacity of computing, communicating, and storage is very limited, which requires WSN protocols need to conserve energy as the main objective of maximizing the network lifetime. An energy-efficient communication protocol LEACH, has been introduced which employs a hierarchical clustering done based on information received by the BS. The BS periodically changes both the cluster membership and the cluster-head (CH) to conserve energy.

The CH collects and aggregates information from sensors in its own cluster and passes on information to the BS. By rotating the cluster-head randomly, energy consumption is expected to be uniformly distributed. However, LEACH possibly chooses too many cluster heads at a time or randomly selects the cluster heads far away from the BS without considering node`s residual energy. As a result, some cluster heads drain their energy early thus reducing the Lifespan of WSN [6].

Cluster Formation algorithm:

1. Create dynamic co-ordinates for nodes.
2. Locate node on above co-ordinates found.
3. Calculate distance of all nodes from a specific node.
4. Store distance in an array dist.
5. Sort dist array and make an array for index storage.
6. Form a cluster based on sorted array of that the dist of node is nearer to the specific node.

6.2 CH Selection

The cluster-based wireless sensor network (WSN) can enhance the whole network lifetime. In each cluster, the cluster head (CH) plays an important role in aggregating and forwarding data sensed by other common nodes. A major challenge in the WSN is the appropriate cluster head selection approach.

Here is a cluster head election algorithm:

```
start T1, expires in rand(0,$\tau$ ) //timer, expires in round 1
start T2, expires in rand($\tau$ ,2$\tau$ ) //timer, expires in round 2
announFirst = (rand(0,1) $\le \gamma$)
CHID = -1 // ID of the cluster head of the node
while T1 NOT expired do
if receive ENC(announcement) AND (CHID = -1) then
```

```
CHID = ID of sender of announcement
end if
end while
// T1 expired
if announFirst AND (CHID = -1) then
broadcast ENC(announcement);
CHID = ID of node itself;
else
broadcast ENC(dummy);
end if
while T2 NOT expired do
if receive ENC(announcement) AND (CHID = -1) then
CHID = ID of sender of announcement
end if
end while
// T2 expired
if (NOT announFirst) AND (CHID = -1) then
broadcast ENC(announcement);
CHID = ID of node itself;
else
broadcast ENC(dummy);
end if
```

6.3 DSDV Protocol

DSDV protocol is based on the Bellman-Ford routing algorithm. It is a proactive protocol and belongs to the table-driven family. Routes between the nodes in the network are always being maintained and updated. Each node in the network maintains a routing table which contains information about how old the route is, the shortest distance as well as the First node on the shortest path to every other node in the WSNs [7].

There are two classified categories for routing updates:

1. full dump update.
2. Incremental update.

Simulation tool:

Software used for the performance analysis of taken protocol is based on NS-2 version 2.27.

NS Simulator based on two languages:

1) An object oriented simulator, written in C++,

2) OTcl (an object oriented extension of Tcl) interpreter.

OTcl used to execute users command scripts. There are two classes' hierarchies: the complied C++ hierarchy and the interpreted OTcl one, with one two one correspondence between them. The complied C++ hierarchy allows us to achieve efficiency in the simulation and faster execution times. This is in particular useful for the detail definition

and operation of protocols. This allows one two reduce packet and event processing time. OTcl script provided by the user, and can define a particular Network Topology, the specific protocols and applications that we wish to stimulate and the form of output that we wish to obtain from the simulator. The

OTcl can make use of the object complied in C++ through an OTcl linkage (done using tclCL) that creates a matching of the OTcl objects for each of the C++. NS is a discrete event simulator, where the advance of time depends on the timing of events which are maintained by a scheduler. An event is an object in the C++ hierarchy with a unique, a scheduled time and the pointer to an object that handles the events.

The schedulers keeps an ordered data structure (there are four, but by default NS use a simple linked- list) with the events to be executed and fires them one by one, invoking the handler of the event. The otcl script used in this simulator is defined in the following manner [7]:

Otcl Script:

```
# Create a simulator object
set ns [new Simulator]
#Open the trace file(s)
set nf [open out.nam w]
nsnamtrace -allnf
#define a 'Finish' procedure proc Finish

global ns nf
nsflush -trace
closenf; #Close the trace file
exec nam out.nam & %Execute nam on the trace file
# (optional)
Exit 0
```

.nam file is generated by.tcl file and we can visualize the network scenario by this.

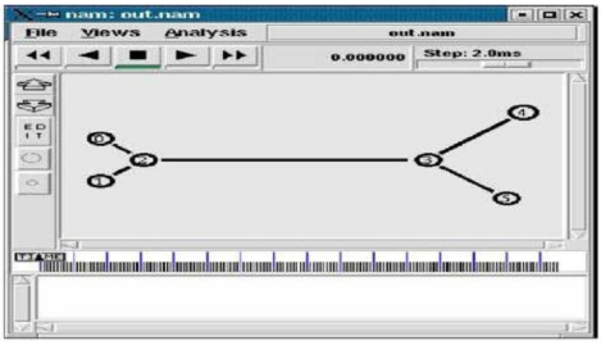


Fig 5.4: Example for creating file nam: nam. out.

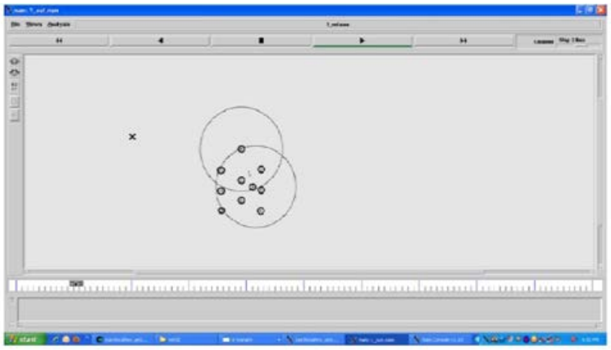
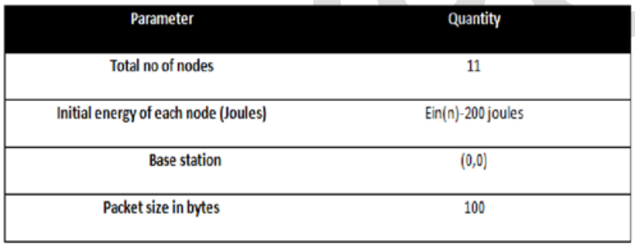


Fig 6.2: Scenario for source and destination variation

7. SIMULATION METRICS & RESULT

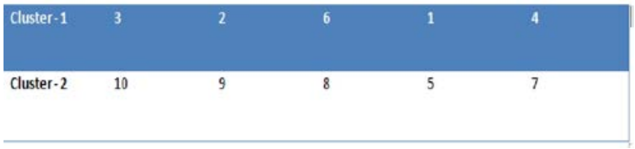
Now in this section, we will see an example of simulation & results of this simulation.



Parameter	Quantity
Total no of nodes	11
Initial energy of each node (Joules)	Ein(n)-200 joules
Base station	(0,0)
Packet size in bytes	100

Fig 6.1 parameters & Quality

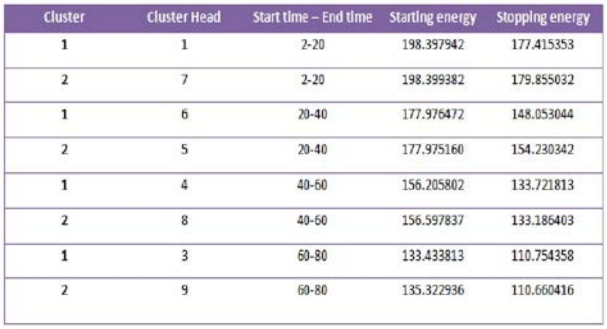
Here 1 node of 11 nodes is especially for BS (Base station) & other nodes are divided for cluster formation in two clusters i.e. cluster-1 & cluster-2 as shown in Fig 6.2.



Cluster-1	3	2	6	1	4
Cluster-2	10	9	8	5	7

Fig 6.2 Clusters & nodes

As cluster head selection is done randomly, ten nodes are divided in 2 cluster as per nearest distance in so there are separate starting & ending times as well as energies for both clusters as shown in fig 6.3.



Cluster	Cluster Head	Start time – End time	Starting energy	Stopping energy
1	1	2-20	198.397942	177.415353
2	7	2-20	198.399382	179.855032
1	6	20-40	177.976472	148.053044
2	5	20-40	177.975160	154.230342
1	4	40-60	156.205802	133.721813
2	8	40-60	156.597837	133.186403
1	3	60-80	133.433813	110.754358
2	9	60-80	135.322936	110.660416

Fig 6.3 Cluster Head rotation & energy changes

From Fig 6.2 & 6.3 we generated two graphs in first graph (fig 6.4) we shown decrease in the energy as per increase in the Time simultaneously. Energy starts with 200 J at initial Time and decreases to 133.43 for next 60s Time.

Change in energy is independent for both clusters.

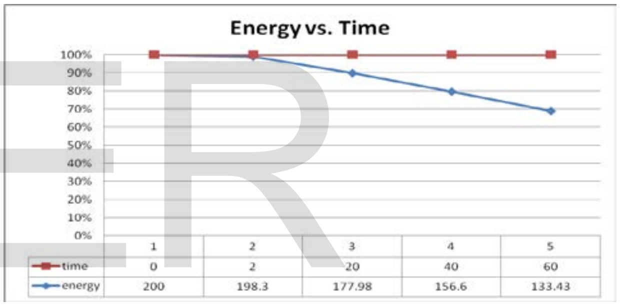


Fig 6.4 Energy vs. Time graph

In Second graph (Fig 6.5), we plotted energy vs. Time from opposite sides to show the simultaneously changes in energy & Time. Here we can notice that as Time increases at the same time energy decreases constantly.

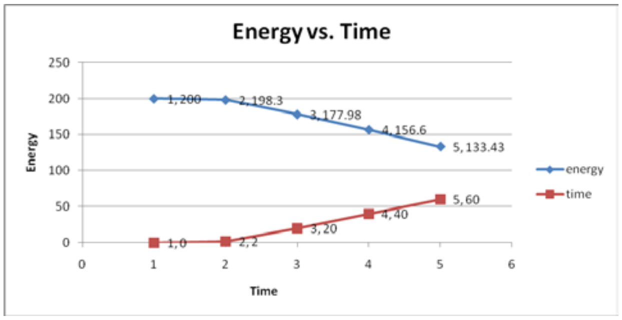


Fig 6.5 Energy vs. Time graph

8. CONCLUSION

This project is created to simulate a clustering protocol called LEACH. This protocol is energy-efficient for a wireless network. The program created was simulation of a network the ranged from 100m x 100m. First, it started with generating just random numbers for coordinates to make the nodes, followed by cluster formation and then adding the codes to for selecting cluster heads and then transmitting and receiving data to show how LEACH works.

ACKNOWLEDGMENT

We would like to thank our guide Prof**. Soumitra S.Das**, for his guidance and support. We will forever remain grateful for the constant support and guidance extended by guide, for the completion of paper.
Also we thank International Journal of Scientific & Engineering Research (IJSER).

REFERENCES

[1]M.Shankar,Dr.M.Sridar,Dr.M.Rajani,"Performance Evaluation of LEACH Protocol in Wireless Network" in International Journal of Scientific & Engineering Research, Volume 3, Issue 1, January-2012 1 ISSN 2229-5518 IJSER.
[2] W. Heinzelman, A. Chandrakasan and H. Balakarishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks" Proceedings of the Hawaaian International
[3]Rajashree.V.Biradar,Dr.S.R.Sawant,Dr.R.R.Mudholkar,Dr.V.C. Patil,"Multihop Routing In Self-Organizing Wireless Sensor Networks,"in Proceedings of the IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011.
[4]Xuxun Liu,"A Survey on Clustering Routing Protocols in Wireless Sensor Networks",sensors ISSN 1424-8220,August 2012.
[5] Lai, W.K.; Fan, C.S.; Lin, L.Y. Arranging cluster sizes and transmission ranges for wireless sensor networks. Inf. Sci. 2012, 183,117131.
[6]K.Ramesh and Dr. K.Somasundaram ,"A COMPARATIVE STUDY OF CLUSTERHEAD SELECTION ALGORITHMS IN WIRELESS SENSOR NETWORKS", International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011
[7]Adel.S.El ashheb,"Performance Routing and performance optimization using multi-hop routing in WSN national Conference on Computer Networks and Communication Systems (CNCS 2012)

AUTHORS PROFILE

- **Prof. Soumitra S.Das** currently pursuing Ph. D in CSE from SATHYABAMA UNIVERSITY, CHENNAI, and he received M.E. Computer Engineering, Currently working as HEAD & Associate Professor with K.J.College of Engineering & Management Research, Pune in the Department of Computer Engineering. Email: **Soumitra_das@yahoo.com**

- **Nitin D.Alety** currently pursuing B.E. in Computer Dept. from K.J.College of Engineering & Management Research, Pune. Email:nitinalety610@gmail.com

- **Kiran H.Hole** currently pursuing B.E. in Computer Dept. from K.J.College of Engineering & Management Research, Pune. Email:k.hole1991@gmail.com

- **Dinesh B.Parmar** currently pursuing B.E. in Computer Dept. from K.J.College of Engineering & Management Research, Pune. Email:dineshparmar181991@gmail.com

- **K.R.Akhil** currently pursuing B.E. in Computer Dept. from K.J.College of Engineering & Management Research, Pune. Email:krakhil44@gmail.com