International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

Cell Hash Routingwith Multiple Paths In Wireless Sensor Networks

Macwan kiran
Master in Information Technology @ PIET
Vadodara , India

Abstract—**This paper focuses on the problem of implementing a distributed hash table (DHT) in wireless ad hoc networks.Scarceness of resources and node mobility turn routing into a challenging problem and therefore, we claim thatbuilding a DHT as an overlay network (like in wired environments) is not the best option. Hence, we present aproof-of-concept DHT, called Cell Hash Routing (CHR), designed from scratch to cope with problems like limitedavailable energy, communication range or node mobility. CHR overcomes these problems, by using position information to organize a DHT of clusters instead of individual nodes. By using position-based routing on top ofthese clusters, CHR is very efficient. Furthermore, its localized routing and its load sharing schemes, make CHRvery scalable in respect to network size and density. For these reasons, we believe that CHR is a simple and yetpowerful adaptation of the DHT concept for wireless ad hoc environments.**

I. INTRODUCTION

P2P Network is the network without the notion of clients or servers but only equalpeer nodes that simultaneously function as both client and servers. Wireless sensornetwork is peer to peer network, most mobile peer to peer approaches deploy an unstructured peer to peer network on top of ad-hoc routing protocol, such as AODV or DSR ,thus they suffer from the limited scalability of ad-hoc routing protocols andthe limited scalability of unstructured network as both layers strongly rely on Flooding. So some scalable routing protocol is needed for wireless sensor network. Inrecent years, wireless sensor networks (WSNs) has gained a lot of attention in bothresearch and application fields. A fundamental goal of researches in WSNs is to reduce the communication operations and prolong the total lifetime of sensor networks.Therefore, researchers are focusing on optimal routing techniques which are one ofthe main factors to reduce consumable energy of sensor and wireless sensor networks.Distributed Hash Tables (DHTs) over WSNs is a new routing paradigm promises several advantages over conventional routing protocols. This paper presents an overviewof using DHTs in WSNs, especially routing techniques using DHTs over WSNs suchas GHT, CSN, CHR, T-DHT, VRR and Scatter Pastry. After that we compare withmethods which used DHTs in WSNs with three main parameters such as Scalability,Energy efficient, and Data storage/lookup Efficiency.

DHT (Distributed Hash Table): It maps Data items, key-value pairs, on node IDs. Key is computed via hash function from string describing data item. Node ID isderived from its Geographical position. Properties of DHT:

- High scalability
- High robustness against frequent peers departures and arrivals
- Self organization: No need for a central server
  1. No single-point failure problem
  2. Higher fault tolerance

Why DHT in peer to peer Network?

- Balanced distribution of data among nodes: Avoid having nodes overloadedwith data.
- Minimum disruption by nodes joining, leaving and failing: Only a small part ofthe network concerned by a change in the set of Participants

1. Consistency

II. CELL HASH ROUTING (CHR)

A. *Working Of CHR*

CHR has [8] focused on the problem of implementing a distributed hash table (DHT) in wireless ad hoc networks. It is designed from scratch to cope with problems like limited available energy, communication range or node mobility. CHR supposed to overcome the limited available energy, communication range and node mobility by using position information on top of clusters of DHTs. CHR has divided the space into equally size cells that the shape of squares, as the grid illustrated in figure 1. Allnodes inside the same cell belong to the same cluster. CHR has used the notion ofcluster as a kind of "super node", where interactions occur at the level of clusters.Since cells are immutable, clusters can receive an identification that does not changewith time. Therefore, nodes can always determine the identification of their cluster,as well as identify other nodes in the same cluster, even in dynamic scenarios, wherethey move around and change from cell to cell. The size of the cells is limited by thecommunication range of the nodes, because a node in a cell must always listen to anya) b) other node either in its own cell or in any adjacent cell. Assumption that nodeshave a communication range of R. so the square size is at most R/ 8. CHR uses arouting scheme based on GPSR combined with a pre-processing algorithm. The hashfunction determines the single cluster that will hold the (key, value) pair. In the caseof a given pair (keyA, valueA), the cluster whose identification equals hash(keyA)will be responsible for storing value A. For instance, consider the ("Bob", 18) pair,where the key "Bob" hashes to 144. In this case, the value 18 should be stored at thecell 144. Formula following shows how to determine the address of a cell in thisscheme. Dx and Dy are the size of the space in the two dimensions, dx and dy arethe sizes of each cell and Lx and Ly

are the coordinates of the center point of the cell(it can also be any other point inside the cell). For instance, this equation is usefulto let a node determine the number of its own cell.

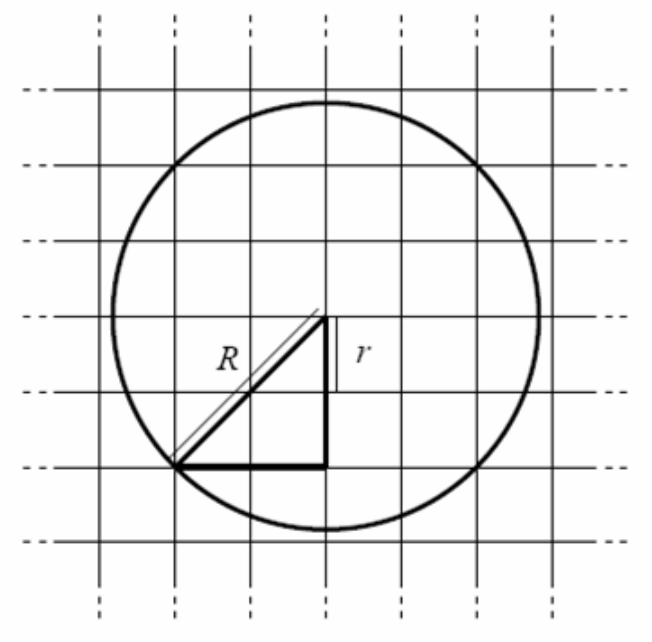


Figure 1:Division of the space into cells of fixed size

By using GPSR to route messages, CHR can follow an approach similar to GHT to resolve empty cells is that some keys may be left without nodes to store them. CHR is more robust, because a single key may be stored at many nodes, allowing the DHT to resist better to abrupt departure of nodes.

B. Comparison

Approch	Scalability	Energy Efficent	Data Lookup Efficiency
GHT	+	+	++
CSN	+	+	++
T-DHT	++	–	+
CHR	+	+	+
VRR	+	+	++

Table 1

C. Problem Statement

- Void cell: [8] it may be possible cell doesn't have any node active in it. So when DHT maps key and value to void cell. It creates problems. And also supposecell has only 1 or 2 nodes active then it may cause data unreachable when thesenodes are going down.
- General Non-Udg model [8]: in more general models, it is possible for anode not to see some of its neighbors in its cell or in some adjacent cell.
- Incorrect Determination of position:[8] in all the papers, we assumed thatnodes can always determine their position exactly, which is actually not thecase. The consequence of this is that a node may consider itself to be in thewrong cell. In a sense, this resumes to the non-UDG model.
- Cluster Induce Disconnection: [8] occasionally, the clustering mechanismmay disconnect the network. This event should be rare, especially in densernetworks, where use of CHR is more worthwhile.
- Fault tolerance requirements:[8] one of the occurrences that CHR shouldtry to avoid as much as possible is the loss of stored (key, value) pairs.
- Network Life Time: maximum network life time.

D. Possible Solution

- Void Cell: When the number of nodes in a cell drops below l, the cell is considered empty. On the contrary, the cell needs to acquire h nodes before it is considered populated (h > l). Note that the value h should be fairly small. As a consequence, knowing h does not require much memory, because, in general, a node will not need to know all the neighbors in its own cell a cell leaving the network delivers its keys to its home cell. An entering cell needs to query its home perimeter to receive its keys. Additionally, it will also receive keys ofempty cells for which it becomes the home cell.
- Network Life Time: Increase the network life time by implementing roundrobin algorithm between three paths between source and destination.
- Fault Tolerance: We can also use multiple paths to provide Fault Tolerance because if one path is fail to transmit packet than we have another path totransmit that packet.

III. IMPLEMENTATION

A. Work Flow

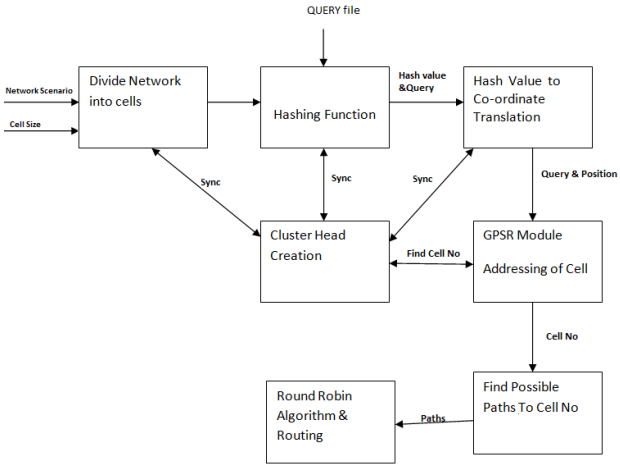


Figure 2: Work Flow

Work flow of implementation is shown as above. Here first we are dividing networkinto cells from the given network scenario as well as simultaneously creating cluster head for each cell. This module is deal with managing cluster head. Now every querypassed through hash function in each node which it passed through. Hash function willgive node no to which query has to send. As GPSR is dealing with geographical location therefore we need to translate hash value to co-ordinate value. After translating itto co-ordinate value, co-ordinates and query are to be sent to the gpsr module.GPSR module find the cell no and it cluster head from the co-ordinates value and routethe query to cell no. But in between gpsr finds all the possible paths from source todestination and Round robin algorithm forward the query packets to each path.

IV. BUILDING CELL HASH ROUTING PROTOCOL

A. Cell Division

We are using GAF(geographic Adaptive Fidelity) protocol for topology control and dividing topology into Cells. the idea is to divide the area into rectangles that are small enough such that any node in one rectangle can communicate with any other node in adjacent rectangle. Node should communicate at maximum radio range. Figure illustrates this relationship for rectangle length r and maximum radio range R.

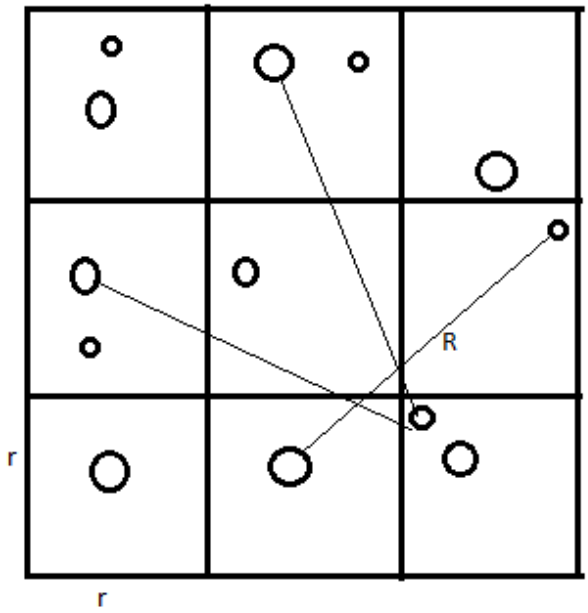


Figure 3: GAF

The distance between two such critical nodesis $\sqrt{r^2 + (2r)^2}$. As this distance has to be smaller than R it follows that r <R/√5.since node are assumed to know their location, they can easily construct such equivalency rectangles. We have integrated GAF module with NS-2 to use it in our TCL script.

B. DHT

We illustrate one possible mapping with a very simple example that assumes a bounded geographical space whose bounds are known by all the nodes. Equation 1 shows how to determine the address of a cell in this scheme. Dx and Dy are the size of the space in the two dimensions; dx and dy are the sizes of each cell and Lx and Ly are the coordinates of the center point of the cell. For instance, this equation is useful to let a node determine the number of its own cell.

1

$$A = [Dx/dx] * [Ly/dy] + [Lx/dx]$$

The reverse correspondence is also useful to allow nodes to perform geographical routing in G. Equations 2 and 3 determine the center point (Lx;Ly) of the cell. C represents the number of columns and is computed as c = [Lx/dx] while % is the remainder of the division. To route to a given cell A, nodes need to determine the center point (Lx;Ly) of the destination cell, before they apply the GPSR routing algorithm. These equations are needed because geographical routing takes place using the center points of the cells (graph G), while the DHT addresses of the cells are only logical. Consider again the "Bob"; 18 pair, where the key hashes to 144. To compute the center (Lx;Ly) of the target cell, a given node would have to replace A by 144 in the Equations 2 and 3.

2

$$Ly = dy([A/c] + 0.5)$$

3

$$Lx = dx([A\%c] + 0.5)$$

V. IMPLEMENTING ROUND ROBIN ALGORITHM

First of all when any packets sequence for the same destination comes to the node, node finds all the neighbor cells and its corresponding cluster head. After it path finding algorithm finds the appropriate three paths in the direction of destination as shown in figure

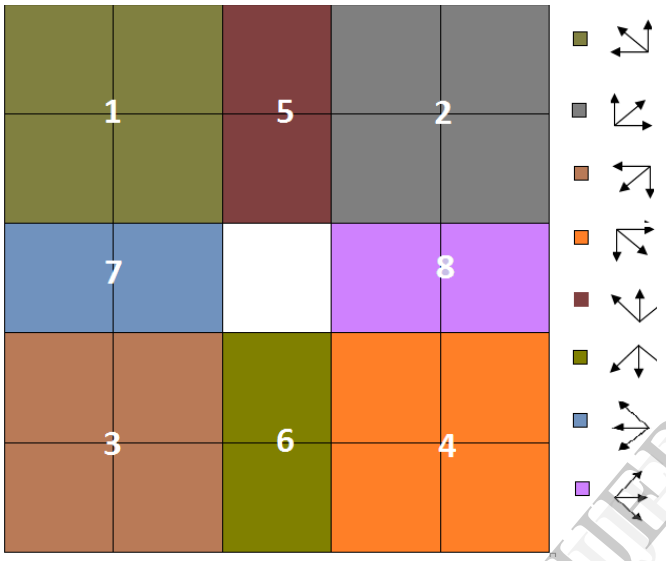


Figure 4: Region Division

.In figure 4 we have shown 8 different regions and also we have shown directions of paths to the destination node of that particular region. Now we have three different paths to each destination node and we are doing round robin over these three paths for sending packets from one source to destination.

VI. RESULTS

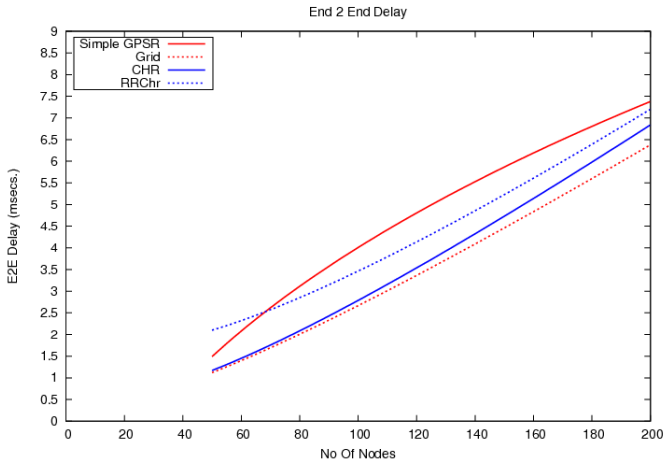


Figure 5:E2E Delay

Analysis: We can see from figure 5 that Clustering reduces End to End Delay but when we applied Hashing on the second implementation than hashing slightly increases End to End Delay because of Hashing Calculation of nodes. After this when we applied Round Robin Algorithm on third implementation than Round Robin Algorithm also increases End to End Delay because of Multiple Paths that all are not shortest paths from source to destination so packets have to travel more than in previous implementation. Overall last implementation improves End to End Delay by 2.43 %.

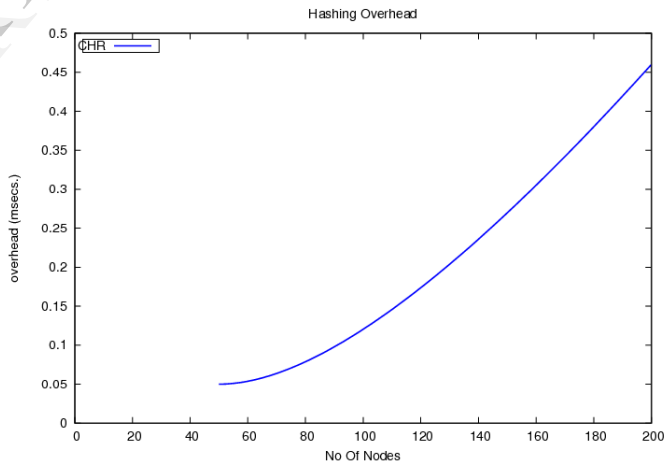


Figure 6: Hashing Overhead

Analysis: Figure 6 is showing the overhead of Hashing. Here we have taken increase in End to End Delay because of Hashing as overhead. It grows exponentiallywith large no of nodes in the network.

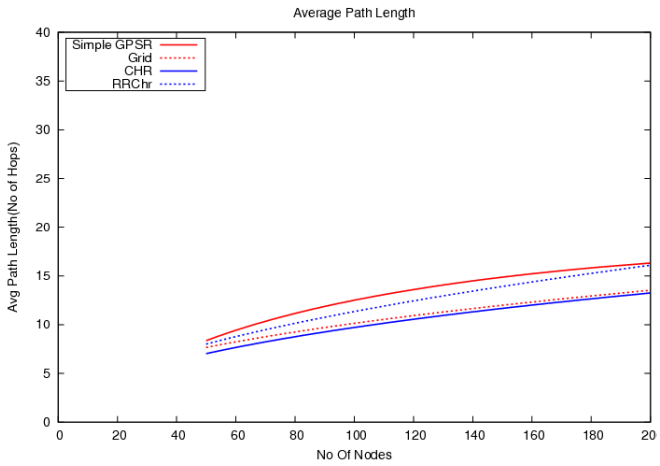


Figure 7: Avg Path Length

Analysis: Figure 7 is showing the average no of hops that packet has to travelto reach destination. As we know clustering always reduces hop count so when weapplied clustering on simple GPSR than it reduces average hop counts but when weapplied Round Robin Multiple paths Algorithm than it slightly increases the hopcounts because of multiple paths. Here proposed implementation improves AveragePath Length by 1.4 % over simple GPSR.

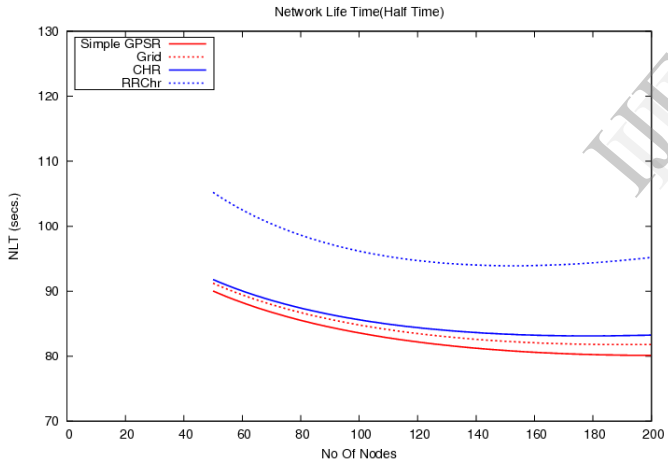


Figure 8: NetworkHalf Time

Analysis: Here we have taken half time as time at which half of nodes will godown;as we can see from the figure 8 that Round Robin Algorithm increases theHalf time of Network by around 18 %.

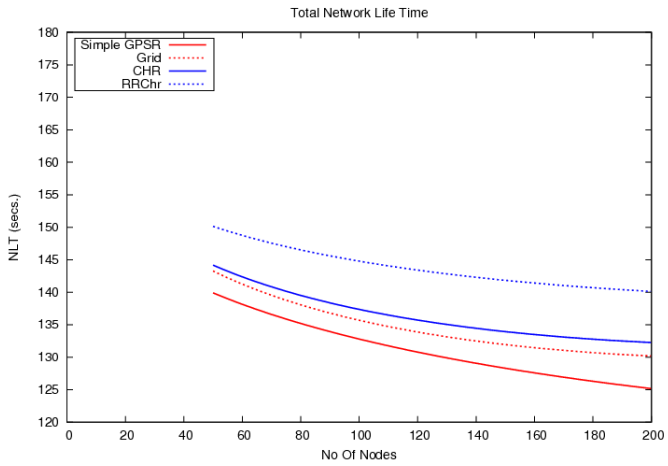


Figure 9: Total Network Life Time

Analysis: Here we have taken Total Life time as time at which all nodes will godown, as we can see from the figure 9that Round Robin Algorithm also increasestotal time of Network by 15 %.

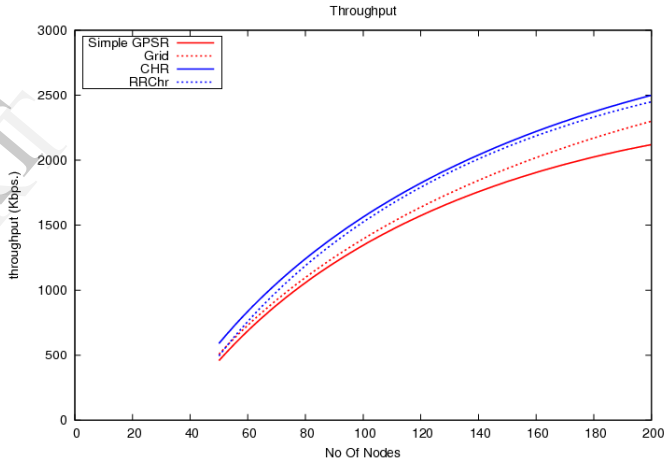


Figure 10: Throughput

Analysis: Throughput is bandwidth of channel which is increasing when we applied clustering and hashing on simple GPSR but when we applied Round Robin Overit than it is slightly lower than third implementation because of longer paths. Proposed implementation improves throughput from 10 to 15 %.

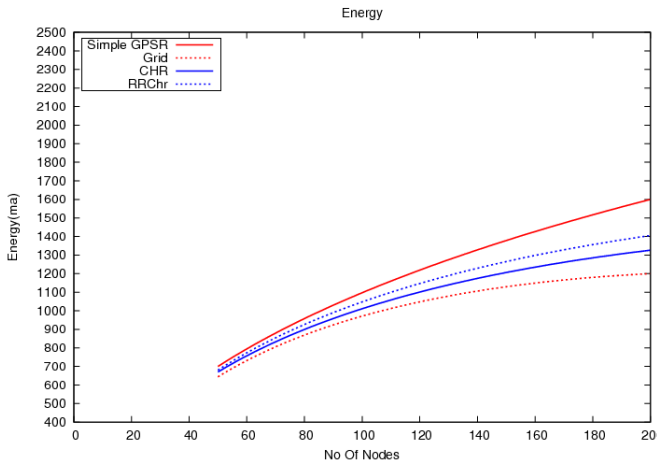
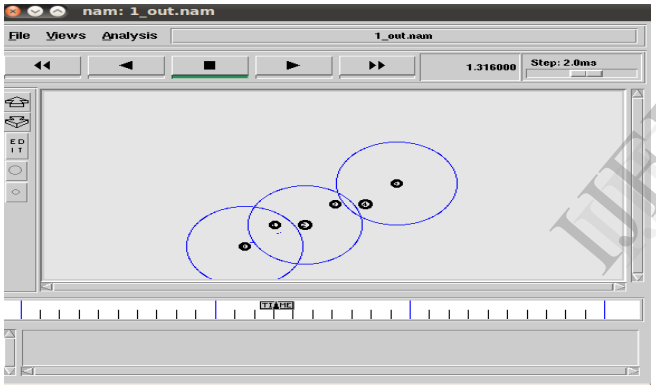
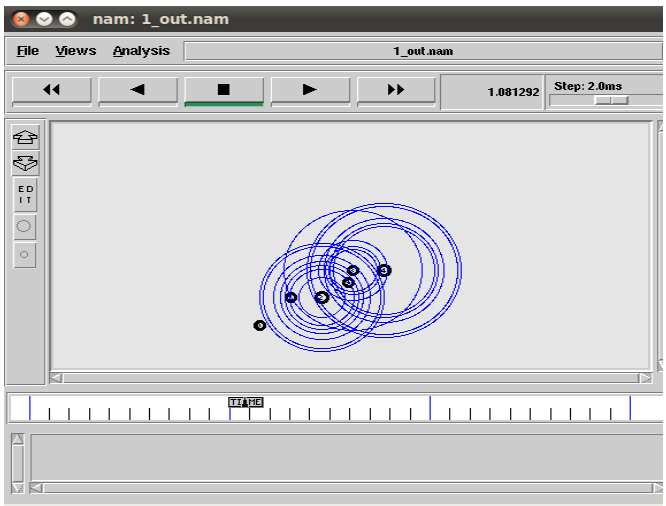


Figure 11: Energy

Analysis:Figure 11 is showing total energy consume in network during network simulation time of 100 seconds. Implementation increases total energy consumption around 5 % over CHR as we have a larger average path length. Because of Larger Average Path Length more nodes are sending and receiving same packet to reach destination in this way they consume more energy.





CONCLUSION

In this dissertation, I have proposed a method to implement Round Robin Algorithm in CHR protocol for wireless sensor network with some assumptions. It finds the multiple paths from source to destination and apply round robin over those paths. So it reduces the load over single path and every path gets a fair chance to transmit packets. Proposed method increases network life time. From the results we can see that goal of thesis - "Distributed Hash Table for Scalable Wireless Sensor Network" is achieved but as an odd effect this proposed method increases End to End Delay. It is to be noted that throughput decreases slightly but it is almost close to CHR implementations.

VII. FUTURE SCOPE

In future, the idea presented over here can be extended by solving the following problems by possible solution mentioned here,

- Dynamic Cell Structure: When the number of nodes in a cell drops below l, thecell is considered empty. On the contrary, the cell needs to acquire h nodes before itis considered populated (h >l). Note that the value h should be fairly small. As aconsequence, knowing h does not require much memory, because, in general, a nodewill not need to know all the neighbors in its own cell. A cell leaving the networkdelivers its keys to its home cell. An entering cell needs to query its home perimeterto receive its keys. Additionally, it will also receive keys of empty cells for which itbecomes the home cell.
- General non-UDG model: In more general models, it is possible for a node not tosee some of its neighbors in its cell or in some adjacent cell. One possible approachto overcome this problem is to create routing tables to reach only the invisible nodesinside the cell or adjacent cells.

REFERENCES

[1] Vinh Vu Thanh, Hung Nguyen Chan, Binh Pham Viet, Thanh Nguyen Huu "A survey of routing using DHTs over Wireless Sensor Networks" in The 6th Inter- national Conference on Information Technology and Applications (ICITA 2009)

[2] Sylvia Ratnasamy and Brad Karp ICIR/ICSI, Berkeley, CA 94704 sylviar, Yin Liand Fang Yu Berkeley EECS, Berkeley, CA 94704 yinli , Deborah Estrin UCLA Comp. Sci., LA, CA 90095,

[3] Ramesh Govindan USC Comp. Sci., LA, CA 90089,Scott Shenker ICIR/ICSI, Berkeley, CA 94704 "GHT: A Geographic Hash Table for DataCentric Storage"

[4] Olaf Landsiedel, Stefan Gotz, Klaus Wehrle Distributed Systems Group RWTH Aachen, Germany _rstname.lastname@cs.rwth-aachen.de "Towards Scalable Mobility in Distributed Hash Tables" in Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06)

[5] Frank Dabek "A Distributed Hash Table" in Submitted to the Department of Electrical Engineering and Computer Science in partial full_llment of the requirements for the degree of Doctor of Philosophy at

the Massachusetts Institute of Technology September 2005 Massachusetts Institute of Technology

[6] Prof Narendra Kumar, Dean DIT School of Engineering and a student of PhD at Manav Bharti University, Solan, Himachal Pradesh (India)," Data Centric Storage in Wireless Sensor Networks"in International Journal of Advances in Engineering Research (IJAER) 2011, Vol. No. 1, Issue No. VI, JUNE

[7] Olaf Landsiedel, Katharina Anna Lehmann, Klaus Wehrle Wilhelm-Schickard Institute for Computer Science University of Tubingen, Germany "T-DHT: Topology-Based Distributed Hash Tables"

[8] Karp, B. and Kung, H.T., Greedy Perimeter Stateless Routing for Wireless Networks, in Proceedings of the Sixth Annual ACM/IEEE International Conferenceon Mobile Computing and Networking (MobiCom 2000), Boston, MA, August, 2000, pp. 243-254.

[9] Tutorial for the Network Simulator ns-2 "http://www.isi.edu/nsnam/ns/tutorial/"