A Data Transfer in Wireless Sensor Networks Using AODV Protocol

B.Manimekala[1], M.Kayalvizhi[2]

[1] Assistant Professor, Department of MCA & Software Systems, V.L.B.Janakiammal College of Arts and Science, Coimbatore, TamilNadu, India

[2] Assistant Professor, Department of MCA & Software Systems, V.L.B.Janakiammal College of Arts and Science, Coimbatore, TamilNadu, India

Abstract

Sensor networks present unique opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military application and environmental monitoring, chemical/biological detection etc. Sensor Network can monitor ambient condition such as temperature, sound, light and others. Information is collected from many sensor devices for further consumer application in the Sensor Network. For selecting a cluster head, k-means algorithm will be used to find the cluster center. For hop to hop packet forwarding, AODV protocol has been used at network layer. All the simulations of the proposed idea will be simulated on Berkeley's ns2 network simulator and the performance of the proposed scheme has been evaluated.

Keywords: Sensor network, AODV, NS2, Cluster.

1. Introduction

1.1 About Sensor Networks

Sensor networks present unique opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military applications, environmental monitoring, chemical/biological detection etc. Microsensor systems enable the reliable monitoring and control of a variety of applications

1.2 Problem Definition

Wireless Sensor Network is one promising application on wireless ad hoc networks. Sensor Network can monitor ambient condition such as temperature, sound, light and others. Information is collected from many sensor devices for further consumer application in the Sensor Network.
The proposed clustering based architecture is similar to the hierarchical fusion architecture. The hierarchical management architecture can be applied to handle numerous sensor nodes. The lower-level nodes are managed and organized by the higher-level nodes by using 20/80 rules. The routing protocols for ad hoc networks can be divided into two categories: table-driven and on-demand routing protocols. The table-driven routing mechanisms discover and maintain routing tables even when no usage of network; in on-demand routing mechanisms, the routes are discovered only when it is need by the source node. These two routing categories have their own merits and demerits. This project a new cluster-based routing protocol that combines the on-demand routing in inter-cluster and the table-driven routing in intra-cluster to be suited for the environment of Sensor Networks.

1.3 About This Research

This project proposes a scheme for Self-Organization Management Protocols of higher-level nodes to contest member nodes with multi-hop form clusters, introduce the "20/80 Rule" for determining the ratio of headers to member nodes. This study implements the proposed management protocols including Clustering Mechanism for constructing cluster headers to solve the problems of clustering and broadcast storm, the suitable protocol to provide low cost communications between clusters.
The Location information of the sensor nodes will be collected from source and the cluster heads were elected/selected based on the locations of the active sensor nodes in network. For selecting a cluster head, k-means algorithm will be used to find the cluster center. The nearby sensor nodes of a cluster-head then will forward their data to sink only via the cluster-head. For hop to hop packet forwarding, AODV or DSDV protocol may be used at Network Layer Protocol. That is to forward a packet from a sensor node to a cluster-head or cluster head to sink; normal routing protocols may be used.
The proposed clustering based application layer

protocol reduces the number of management nodes in a large-scale sensor network. All the simulations of the proposed idea will be simulated on ns2 network simulator and the performance of the proposed scheme will be evaluated.

2. Sensor Network

A sensor network comprises of sensor nodes and a base station. Each sensor node is battery powered and equipped with:

1. Integrated sensors
2. Data processing capabilities
3. Short-range radio communications

2.1 Data Fusion in Sensor Network

In sensor networks, thousands of sensor nodes collectively monitor an area. These large sensor networks generate a substantial amount of data, yet the sensor nodes often have limited resources, such as computation power, memory, storage, communication, and most importantly, battery energy. The process wherein, sensors detect an event, and the data related to the event is eventually fused at the sink via multiple levels of fusion en route, is called a "round" of aggregation. Typically, the sink is distant from the area where the sensor nodes reside. The sensor data has to be ultimately relayed to the sink via multiple sensor node relays.

The large scale of sensor networks and the resource constraints make it an important challenge to design and develop efficient information processing and aggregation techniques to make effective use of the data.

Given a query, it may be unnecessary and inefficient to return all raw data collected from each sensor—instead, information should be processed and aggregated within the network and only processed and aggregated information is returned.

In such a setting, certain nodes in the sensor network, called aggregators, collect the raw information from the sensors, process it locally, and reply to the aggregate queries of a remote user.

Each sensor node has to decide on when to begin the process of fusion and how long to wait before the end of the fusion. Intuitively, the longer the time that the sensor node which performs the fusion waits, the larger the number of reports it will receive.

In this setting, we focus on stealthy attacks, where the attacker's goal is to make the Base Station accept false aggregation results, which are significantly different from the true results determined by the measured values.

2.2 Routing in Sensor Networks

There are a few inherent limitations of wireless or wired media such as low bandwidth, error prone transmissions, need for collision free channel access etc. These wireless nodes also have only a limited amount of energy available to them, since they derive energy from a personal battery and not from a constant power supply. Furthermore, since these sensor nodes are deployed in places where it is difficult to replace the nodes or their batteries, it is desirable to increase the longevity of the network. Also preferably all the nodes should die together so that we can replace all the nodes simultaneously in the whole area. Finding individual dead nodes and then replacing those nodes selectively would require pre-planned deployment and eliminate some advantages of these networks. Thus, the protocols designed for these networks must strategically distribute the dissipation of energy, which also increases the average life of the overall system.

2.3 The Proposed Network Architecture

The proposed mechanism reduces the number of management nodes in a large-scale sensor network. One of the best methods is to develop hierarchical architecture and apply the 20/80 rule [1].

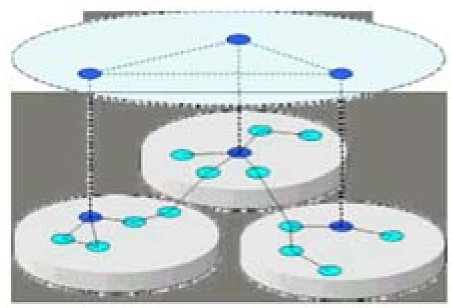

Fig 1: The Architecture of Clustering Network

Construct the intra-cluster routing with table-driven routing mechanisms similar to destination sequence distance vector (DSDV). The source node broadcasts a route query packet to nodes until the border nodes in adjacent clusters around the local cluster receive it. Therefore, the intra-cluster routing limits the broadcast range within one distance hop distance of the cluster to lower the control overhead and reduce the interference of the shared media. Figure 5 shows how the Intra-cluster routing builds.

Also, the proposed protocol acquires the inter-cluster routing by exchanging the relationship between clusters. To perform the inter-cluster route discovery when the route is demanded can reduce the overhead of building and maintaining inter-cluster routing. When the demand for inter- cluster

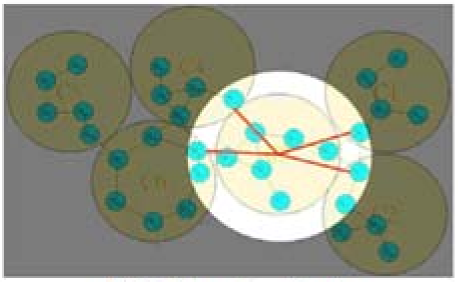

Fig 2: Intra-cluster Routing

route occurs, the source node sends the inter-cluster route request packet (RREQ) in unicast mode to the border nodes to acquire the adjacent cluster's intra-cluster routing information.

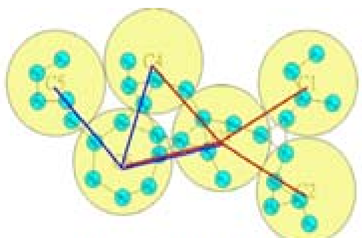

Fig 3: Inter-cluster Routing

After constructing the clustering network, this previous work proposes cluster-based routing algorithm into two aspects: intra-cluster routing and inter-cluster routing [1].

2.4 Classification of Sensor Networks

Sensor networks can be classified into two types based on their mode of operation or functionality and the type of target applications [15] [16].

Proactive Networks: In this scheme the nodes periodically switch on their sensors and transmitters, sense the environment and transmit the data of interest. Thus, they provide a snapshot of the relevant parameters at regular intervals and are well suited for applications that require periodic data monitoring.

Reactive Networks: In this scheme the nodes react immediately to sudden and drastic changes in the value of a sensed attribute and are well suited for time critical applications.

Once we have a network, we have to come up with protocols which efficiently route data from the nodes to the users, preferably using a suitable MAC (Medium Access Control) sub-layer protocol to avoid collisions.

2.5 Proactive Network Protocol

At each cluster change time, once the cluster-heads are decided, the cluster-head broadcasts the following parameters:

Report Time (T_R), is the time period between successive reports sent by a node.

Attributes (A), is a set of physical parameters which the user is interested in obtaining data about.

At every report time, the cluster members sense the parameters specified in the attributes and send the data to the cluster-head. The cluster-head aggregates this data and sends it to the base station or a higher level cluster-head, as the case may be. This ensures that the user has a complete picture of the entire area covered by the network.

Since the nodes switch off their sensors and transmitters at all times except the report times, the energy of the network is conserved. At every cluster change time, T_R and A are transmitted afresh and thus can be changed. By changing A and T_R, the user can decide what parameters to sense and how often to sense them. Also, different clusters can sense different attributes for different T_R. This scheme, however, has an important drawback. Because of the periodicity with which the data is sensed, it is possible that time critical data may reach the user only after the report time making this scheme ineffective for time-critical data sensing applications.

LEACH (Low-Energy Adaptive Clustering Hierarchy) is a family of protocols developed in [14]. LEACH is a good approximation of a proactive network protocol, with some minor differences.

Once the clusters are formed, the cluster heads broadcast a TDMA schedule giving the order in which the cluster members can transmit their data. The total time required to complete this schedule is called the frame time T_F. Every node in the cluster has its own slot in the frame, during which it transmits data to the cluster head. When the last node in the schedule has transmitted its data, the schedule is repeated.

The report time T_R discussed earlier is equivalent to the frame time T_F in LEACH. However T_F is not broadcasted by the cluster head but is derived from the TDMA schedule and hence is not under user control. Also, the attributes are predetermined and not changed after initial installation.

In a new network protocol called TEEN [15] (Threshold sensitive Energy Efficient sensor Network protocol) has been developed that is targeted at reactive networks and is the first protocol developed for reactive networks.

3. Simulation of Proposed Sensor Network

This section gives a brief overview of each simulator. Its aim is to summarize the different implementation approaches of each simulator. The way a new algorithm is integrated can be pretty different from one simulator to another. The various simulators commonly available are: OPNET, NS-2,

OPNET Modeler is a powerful network simulator developed by OPNET. It can simulate all kinds of wired networks, and an 802.11 compliant MAC layer implementation is also provided. Although OPNET is rather intended for companies to diagnose or reorganize their network, it is possible to implement one's own algorithm by reusing a lot of existing components. Most part of the deployment is made through a hierarchical graphic user interface. Basically, the deployment process goes through the following phases.

First step it to choose and configure the node models (i.e. types) that is to be used in the simulations -for example a wireless node, a workstation, a firewall, a router, a web server, etc. Then build and organize the network by connecting the different entities. The last step consists in selecting the statistics needed to collect during the simulations.

The difficulty with OPNET Modeler is to build this state machine for each level of the protocol stack. It can be difficult to abstract such a state machine starting from a pseudo-coded algorithm.

But anyway, state machines are the most practical input for discrete simulators. In summary, it is possible to reuse a lot of existing components (MAC layer, transceivers, links, etc.) improving the deployment process. But on the other hand, any new feature must be described as a finite state machine which can be difficult to debug, extend and validate.

NS-2 is a discrete event network simulator that has begun in 1989 as a variant of the REAL network simulator. Initially intended for wired networks, the Monarch Group at CMU have extended NS-2 to support wireless networking such as MANET and wireless LANs as well. Most MANET routing protocols are available for NS-2, as well as an 802.11 MAC layer implementation.

NS-2's code source is split between C++ for its core engine and OTcl, an object oriented version of TCL for configuration and simulation scripts. The combination of the two languages offers an interesting compromise between performance and ease of use.

Implementation and simulation under NS-2 consists of 4 steps:

- Implementing the protocol by adding a combination of C++ and OTcl code to NS-2's source base;
- Describing the simulation in an OTcl script;
- Running the simulation and
- Analyzing the generated trace files.

Implementing a new protocol in NS-2 typically requires adding C++ code for the protocol's functionality, as well as updating key NS-2 OTcl configuration files in order for NS -2 to recognize the new protocol and its default parameters. The C++ code also describes which parameters and methods are to be made available for OTcl scripting. The NS-2 architecture follows closely the OSI model. We have adapted the implementation of flooding provided in NS-2 in the context of diffusion in sensor networks.

An agent in NS-2 terminology represents an endpoint where network packets are constructed, processed or consumed. Such an Agent was implemented at the Application layer for the broadcast source, and the simulation trace was collected at the MAC layer. Some disadvantages of NS-2 stem from its open source nature. First, documentation is often limited and out of date with the current release of the simulator. Fortunately, most problems may be solved by consulting the highly dynamic newsgroups and browsing the source code. Then code consistency is lacking at times in the code base and across releases. Finally, there is a lack of tools to describe simulation scenarios and analyze or visualize simulation trace files. These tools are often written with scripting languages. The lack of generalized analysis tools may lead to different people measuring different values for the same metric names.

The learning curve for NS-2 is steep and debugging is difficult due to the dual C++/OTcl nature of the simulator. A more troublesome limitation of NS-2 is its large memory footprint and its lack of scalability as soon as simulations of a few hundred to a few thousand of nodes are undertaken.

The following shows specifications of Proposed Simulation Environment

The following hardware is used to simulate the algorithm. The Processor with Pentium IV, 1.7GHz, and Memory: 128 MB and Video card: AGP with 32 Mb RAM. The following software used during developing the system, Operating System: Red Hat Linux 9, Simulator: NS2 version 2.28, Graph: X-Graph, Visualization: NAM (Network animator), Programming Language: TCL Shell Scripting.

3.1 The Working of TCL and NS2

An NS simulation script starts with making a Simulator object instance.

set ns [new Simulator]: generates an NS simulator object instance, and assigns it to variable ns. The line does is the following:

–Initialize the packet format (ignore this for now)

–Create a scheduler (default is calendar scheduler)

–Select the default address format (ignore this for now)

The "Simulator" object has member functions that do the following:

–Create compound objects such as nodes and links (described later)

–Connect network component objects created (ex. attach-agent)

–Set network component parameters (mostly for compound objects)
–Create connections between agents (ex. make connection between a "tcp" and "sink")
–Specify NAM display options

Most of member functions are for simulation setup (referred to as plumbing functions in the Overview section) and scheduling, however some of them are for the NAM display. The "Simulator" object member function implementations are located in the "ns-2/tcl/lib/ns-lib.tcl"file.

3.2 Proposed Clustering Based Algorithm

K-Means algorithm is very popular for data clustering. The proposed data transfer protocol K-Means algorithm is used to cluster the nodes with respect to their locations in the MANET and select a central node in each cluster to make it as a cluster head [11].

1. From sink, resolve the locations of all the nodes in the network.
2. For this use a service like GPS or we can resolve them by using the agents in all the nodes which will provide the information on request. For the second approach simple flooding can be used for initial location discovery. (in this simulation, GPS was assumed)
3. Then Select k Center in the problem space (it can be random).
4. Partition the data into k clusters by grouping points that are closest to those k centers.
5. Use the mean of these k clusters to find new centers.
6. Repeat steps 3 and 5 until centers do not change.
7. Find the nearby central nodes from the calculated cluster centers.
8. Make the central nodes as a cluster head.
9. Now, the Cluster heads will announce its presence to all its neighbors.
10. The neighboring nodes in turn will forward the data to sink via the cluster-head.
11. The sink will periodically update the cluster-heads.
12. From sink, resolve the locations of all the nodes in the network.
13. For this we can use a service like GPS or we can resolve them by using the agents in all the nodes which will provide the information on request. For the second approach simple flooding can be used for initial location discovery. (in this simulation, GPS was assumed)
14. Then Select k Center in the problem space (it can be random).
15. Partition the data into k clusters by grouping points that are closest to those k centers.
16. Use the mean of these k clusters to find new centers.
17. Repeat steps 3 and 5 until centers do not change.
18. Find the nearby central nodes from the calculated cluster centers.
19. Make the central nodes as a cluster head.
20. Now, the Cluster heads will announce its presence to all its neighbors.
21. The neighboring nodes in turn will forward the data to sink via the cluster-head.
22. The sink will periodically update the cluster-heads.

The purpose of study is to experiment with various kinds of simulations on NS2 to understand the behaviors of the algorithms under consideration. The following list shows different kinds of simulation parameters used in the proposed data transfer protocol simulation.

3.3 Simulation Results and Analysis

For the purpose of this study, we have experimented with various kinds of simulations on NS2 to understand the behaviors of the algorithms under consideration. The following list shows different kinds of simulation parameters used in the proposed data transfer protocol simulation.
The General Network Topology Parameters are,
Channel Type: Wireless Channel
Radio-Propagation Model: TwoRayGround
Network Interface Type: *WirelessPhy*
MAC type: Mac/802_11
Interface Queue Type: DropTail/PriQueue
Link Layer Type: LL
Antenna Model: Omni Antenna
Queue Length: 50
Network Layer Protocol: AODV
Size of the Topography: 700 X 500
The Constant Simulation Parameters are,
Active Data Senders: 75% Sensor Nodes
Sensor Data Size: 64 Bytes
Fused Data Size: 512
Sensor Data Interval: 1 Data Packet per seconds
Channel Error Rate: 0.15
Total Simulation Time: 10
The Variable Simulation Parameter has Total Sensor Nodes : 50, 75, 100, 125, and 150
To summarize the different implementation approaches of each simulator. The way a new algorithm is integrated can be pretty different from one simulator to another [10].
NS-2 is a discrete event network simulator that has begun in 1989 as a variant of the REAL network simulator. NS-2's code source is split between C++ for its core engine and OTcl, an object oriented version of TCL for configuration and simulation scripts. The combination of the two languages offers an interesting compromise between performance and

ease of use. Implementation and simulation under NS-2 consists of 4 steps:

– Implementing the protocol by adding a combination of C++ and OTcl code to NS-2's source base;

– Describing the simulation in an OTcl script;

– Running the simulation and

– Analyzing the generated trace files.

Algorithm I – Normal Sensor Network

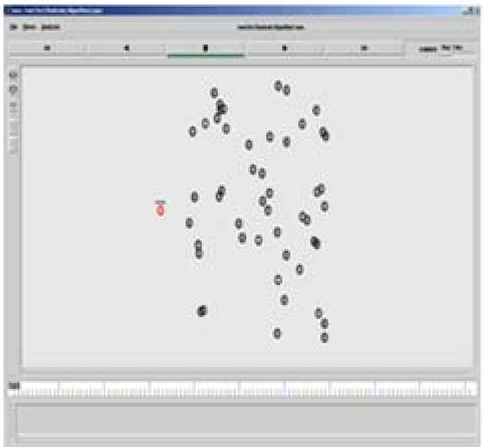

Fig 4: A Sample Sensor Network Topology

All the Sensor Nodes sending Data to the Sink at the periodic intervals.

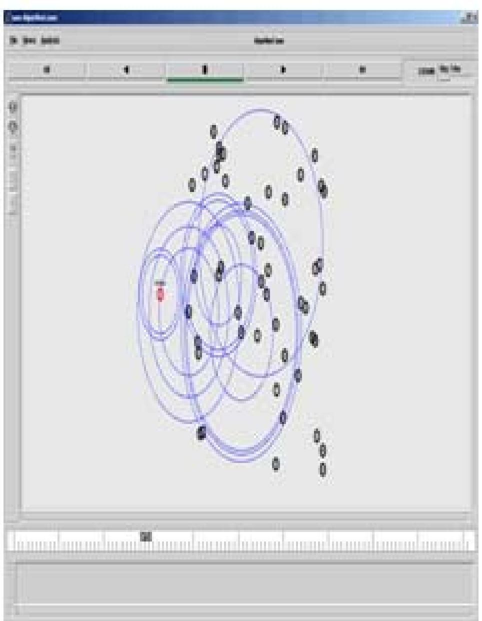

Fig 5: The 50 Node Sensor Network

The Initial Message Broadcast for Route Discovery

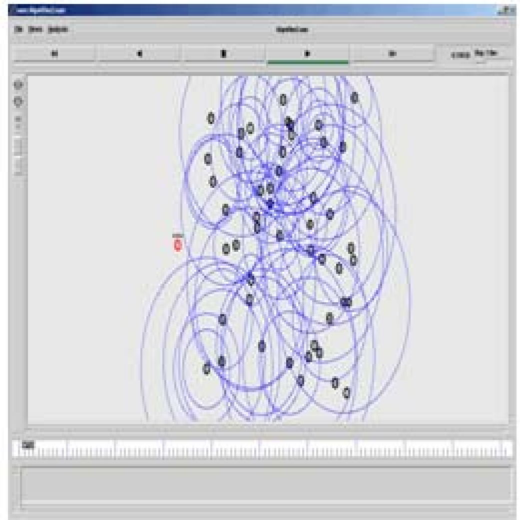

Fig 6: Initial Message Broadcast for Route Discovery

The Clustered Sensor Network, each cluster member is shown in different color. The Sink node (Left-Middle) is shown as red circle.

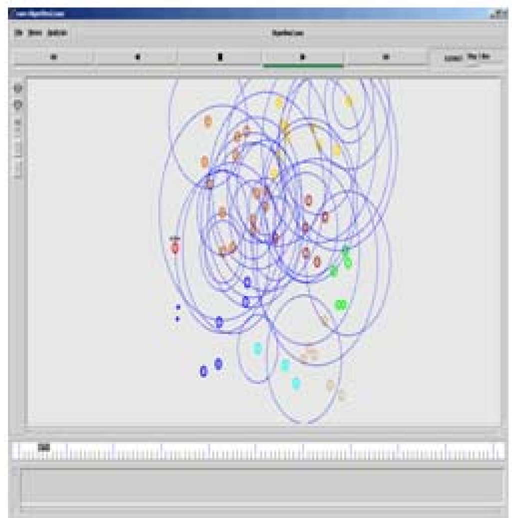

Fig 7: The Clustered Sensor Network

Marking Clusters and Cluster Centres.

The Calculated Cluster Centres

215 174, 413 92, 237 471, 513 383, 567 103, 492 599, 646 282

The Sensor Network with 7 Selected Cluster Heads

The Cluster Heads are shown as marked red Circles (double lines)

The Sink node (Left-Middle) is shown as red circle (single line).

The node Locations nearby the centres

Node 5: (186, 188), Node 39: (434, 93), Node 27: (198, 428), Node 41: (509, 396), Node 11: (543, 114), Node 37: (540, 614), Node 28: (617, 299)

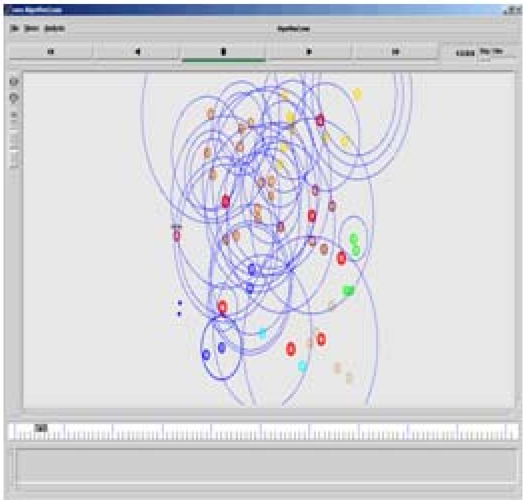

Fig 8: Sensor Network with Selected Cluster Heads

The Cluster Head is forwarding the collected data to sink

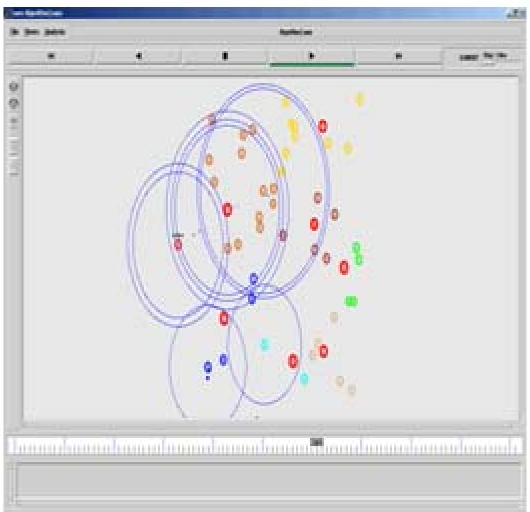

Fig 9: Cluster Head is forwarding the collected Data to Sink

The Final Terminal Window output
Total Sent Bytes: 21056.0
Total Received Bytes: 20928
The Delivery Ratio: 99.392097264437695

3.4 Comparative Results

Metrics considered for evaluation, the following metrics were considered for evaluating the flooding algorithms: Packet Delivery Ratio and Normalized Routing Load.
The following table shows the Packet Delivery Ratio of the two algorithms with respect to different number of nodes.

Sl No	Number of Nodes	Packet Delivery Ratio	
		Normal Method (Algorithm I)	Clustering Based Method (Algorithm II)
1	50	99.25	99.80
2	75	90.87	93.23
3	100	83.98	86.66
4	125	72.67	80.45
5	150	60.32	75.34

Table 1: Number_of_SensorNodes vs. Packet Delivery Ratio

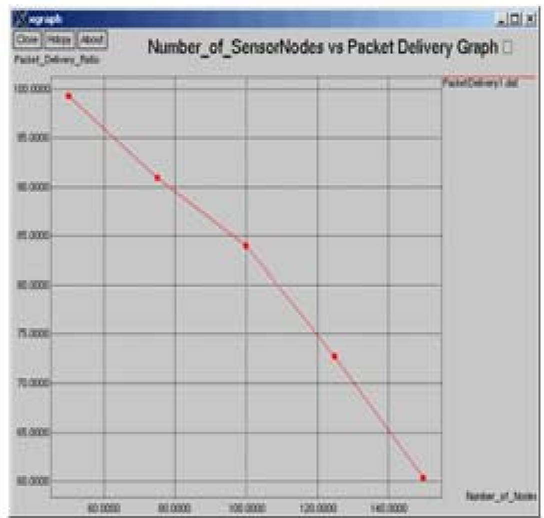

Fig 10: Packet Delivery Ratio of Algorithm I

The Packet Delivery Ratio Graph - Algorithm I

Fig 10: Packet Delivery Ratio of Algorithm I

The Packet Delivery Ratio Graph - Algorithm II

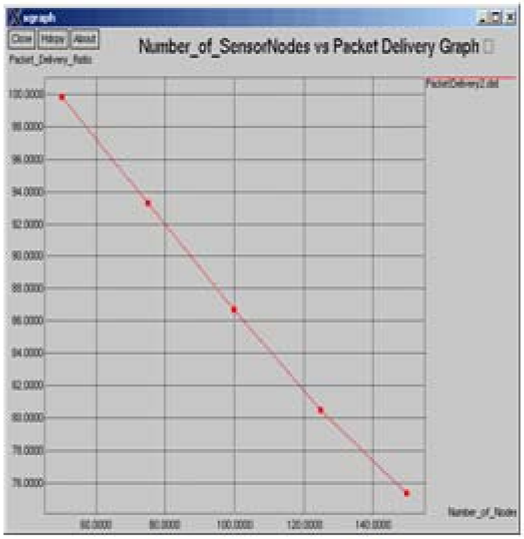

Fig 11: Packet Delivery Ratio of Algorithm II

The following line chart shows the Packet Delivery Ratio of the two algorithms with respect to different number of the nodes.

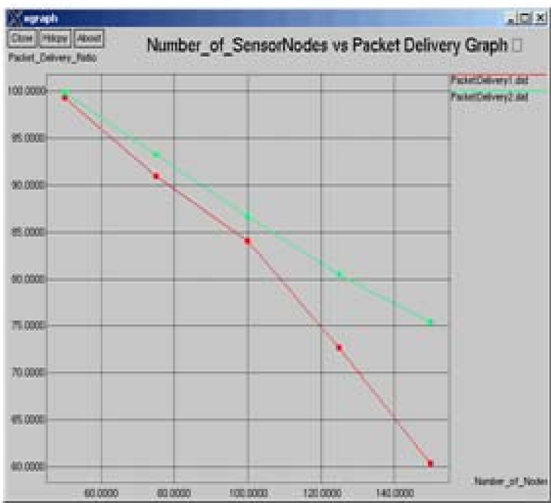

Fig 11: The Combined Packet Delivery Ratio Graph

Normalized Routing Load, the following table shows the normalized routing load of the two algorithms with respect to different number of nodes.

Sl No	Number of Nodes	Routing Load	
		Normal Method (Algorithm I)	Clustering Based Method (Algorithm II)
1	50	0.389	0.263
2	75	0.288	0.198
3	100	0.210	0.132
4	125	0.174	0.090
5	150	0.120	0.075

Table 2: Number_of_SensorNodes vs. Routing_Load

Normalized Routing Load Charts for Number_of_SensorNodes vs. Routing_Load Graph – Algorithm I

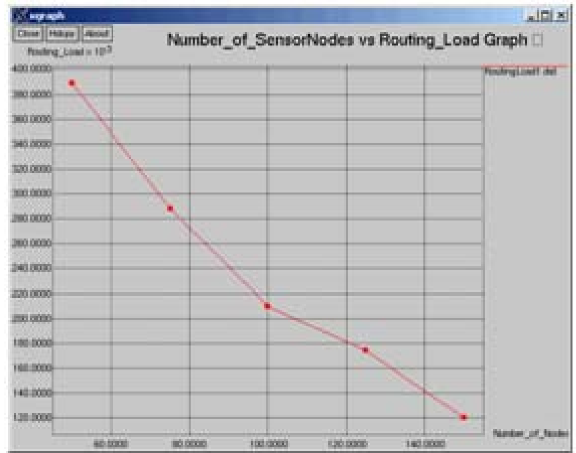

Fig 12: Routing Load of Algorithm I

Number_of_SensorNodes vs. Routing_Load Graph – Algorithm II

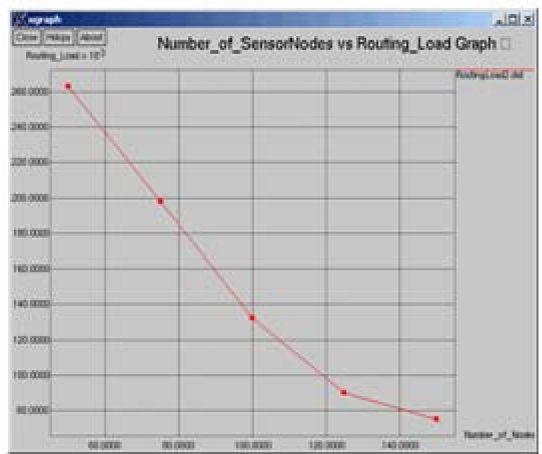

Fig13: Routing Load of Algorithm II

Number_of_SensorNodes vs. Routing_Load - Combined Graph, shows the normalized routing load of the two algorithms with respect to different number of nodes.

Fig 14: Normalized Routing Load Chart

As shown in the previous tables and graphs, the proposed algorithm performed well and provided good packet delivery with minimum routing load.

4. Conclusion and Scope for Further Enhancements

The main scope of this research is to device a new clustering based data transfer protocol for sensor network for efficient data transfer in sensor network. The proposed clustering based data transfer protocol has been successfully implemented and evaluated under Berkeley's Network Simulator ns2.

The performance of the network was tested with different simulation parameters and the simulation was repeated for different number of sensor nodes in the network. According to the trace analysis, the arrived results were significant and more comparable. While comparing the proposed method with a normal method, the proposed data transfer mechanism provided very good packet delivery ratio with very low routing load. The graphs and tables shows the enhancement in performance while using the proposed method.

In this work, the classic k-mean clustering algorithm was used to cluster the mobile nodes. Since the k-means algorithm has some draw backs and produce wrong clusters if there were lot of outliers in the location data, the future works may address more suitable algorithm for clustering the nodes in the mobile scenario.

In this implementation the numbers of clusters were decided with respect to the model scenario at hand. But in future works, one may address the possibility of deciding the number of clusters by some indirect means. In the proposed algorithm, for getting the location information, the same old flooding was used and for further message passing, the proposed clustering based method was used. Future works may address the possibilities of removing the classical flooding phase which is used to discover location information.

The whole experiments were done on Berkeley's Network Simulator. The future works may address the issues for real implementation which may involve real GPS for resolving location information.

Sensor networks present unprecedented opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military applications, environmental monitoring, chemical/biological detection etc. So the future works may address the possibilities of enhancements of the proposed data transfer protocol for specific application.

References

[1] Yao-Chung Chang, Zhi-Sheng Lin and JiannLiang Chen, "Cluster Based Self-Organization Management Protocols for Wireless Sensor Networks", IEEE Transactions on Consumer Electronics, FEBRUARY 2006.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols forWireless Microsensor Networks," In Proceedings Hawaaian International Conference on Systems Science, January 2000

[3] J. Hill, R. Szewczyk, A.Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Network Sensors," In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, November 2000.

[4] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," In Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), August 2000.

[5] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," Technical Report 01-750, University of Southern California, November 2001.

[6] Intel Research Oregon. Heterogeneous sensor networks. Technical report, Intel Corporation, 2003. WebPage.http://www.intel.com/research/exploratory/heterogeneous.htm.

[7] Brad Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," In Proc. ACM/IEEE MobiCom, August 2000.

[8] H. Suzuki and F. A. Tobagi, "Fast Bandwidth Reservation Scheme with Multi link Multi path routing in ATM networks," In Proceedings of IEEE Infocom, 1992.

[9] CITRIS, Center for Information Technology Research in the Interest of Society. Web Page. http://www.citris-uc.org

[10] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols". In Proc. Of 4th Annual ACM/IEEE International Conference on Mobile Computing(MOBICOM), ACM, October 1998.

[11] Elizabeth M. Royer, and C.-K. Toh. "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," In IEEE Personal Communications Magazine, April 1999.

[12] Jeremy Nelson and Deborah Estrin. "An Address-Free Architecture for Dynamic Sensor Networks," Technical Report 00-724, Computer Science Department, USC, January 2000.

[13] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Wireless Networks," In Proc. Of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM),1999.

[14] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "uAMPS ns Code Extensions," http: // www-mtl.mit.edu /research/ icsystems /uamps/leach

[15] A. Manjeshwar and D. P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," In Proc. Of 15th International Parallel and

Distributed Processing Symposium (IPDPS '01) Workshops, 2001.
[16] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," In Proc. Of International Parallel and Distributed Processing Symposium (IPDPS '02) Workshops, 2002.
[16] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," In Proc. Of Hawaiian International Conference on Systems Science, January 2000.
[17] A. Manjeshwar, M.S. Thesis "Energy Efficient Routing Protocols with Comprehensive Information Retrieval for Wireless Sensor Networks", 2001.

B.Manimekala received her B.Sc degree in Computer Science, Master degree (MCA) in Computer Applications and M.Phil in Computer Science in 2003, 2006 and 2010 respectively from Bharathiar University and Periyar University. Previosly she had worked as Assistant Professor inDepartment of Computer Applications, Dr.N.G.P.Arts and Science College, Coimbatore during 2007 to 2010 and she had worked as Assistant professor in Department of Computer Science, Kovai Kalaimagal College Of Arts and Science, Coimbatore during 2010 – 2011. Currently she is working as a Assistant to Professor, Department of Software Systems, V.L.B. Janakiammal College of Arts and Science, Coimbatore. Her area of interest includes Networking, DBMS, and Software Testing. Her career objective is seeking a position to inspire and guide the students to acquire knowledge, develop skills and a positive attitude, which will enhance their personality providing self-confidence to face the competitive world.

M.Kayalvizhi MCA., M.Phil., She is a Research Scholar in Bharathiar University,Coimbatore, Tamilnadu, India. She is working as a Assistant Professor, in VLB Janakiammal Arts and Science College, Coimbatore. she has 5 years of experience in teaching line 4 years of experience in research.she conducted International, National Conference and she presented paper in International, National Conference and Journals. Her Interest areas are Networking Image Processing, Current research project sensor network.