An Enhanced Data Collection for Wireless Sensor Network Using Topology Routing Tree

[I] V.Priya., [II] S.Nivas
[I]Research Scholar, Bharathiar University, Coimbatore, [II] Assistant Professor, CS,
[I, II] Dept. of Computer Science, Maharaja Co-Education College of Arts and Science,
Perundurai, Erode – 638052.
[I] Email id: vpriyainfotech@gmail.com
[I] Contact No: 8344412152
[II] Email id: nivasmaharaja@gmail.com
[II] Contact No:9894376769

Abstract— this paper describes the basic idea about the different methods of data collection in WSN. In many applications of wireless sensor networks, approximate data collection is a wise choice due to the constraints in communication bandwidth and energy budget. Many existing techniques have power to manage with the issues like energy consumption, packet collision, retransmission, delay etc. For quick data collection, schemes are required to be scheduled in effective manner. One of the good techniques is BFS. It provides periodic query scheduling for data aggregation with minimum delay under various wireless interference models. Given a set of periodic aggregation queries, each query has its own period p_i and the subset of source nodes S_i containing the data. Time scheduling on a single frequency channel with the aim of minimizing the number of time slots required (schedule length) to complete a convergecast is considered. Next, scheduling with transmission power control is combined to mitigate the effects of interference, and show that while power control helps in reducing the schedule length under a single frequency, scheduling transmissions using multiple frequencies is more efficient. Lower bounds on the schedule length are given when interference is completely eliminated, and propose algorithms that achieve these bounds. Then, the data collection rate no longer remains limited by interference but by the topology of the routing tree.

Keywords— Wireless sensor network, data collection, energy, aggregation, scheduling, transmission

INTRODUCTION

Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life, from environmental monitoring and conservation, to manufacturing and business asset management, to automation in the transportation and health care industries. The design, implementation, and operation of a sensor network requires the confluence of many disciplines, including signal processing, networking and protocols, embedded systems, information management and distributed algorithms. Such networks are often deployed in resource-constrained environments, for instance with battery operated nodes running un-tethered.

These constraints dictate that sensor network problems are best approached in a hostile manner, by jointly considering the physical, networking, and application layers and making major design tradeoffs across the layers. Advances in wireless networking, micro-fabrication and integration (for examples, sensors and actuators manufactured using micro-electromechanical system technology, or MEMS), and embedded microprocessors have enabled a new generation of massive-scale sensor networks suitable for a range of commercial and military applications.

The technology promises to revolutionize the way we live, work, and interact with the physical environment. In a typical sensor network, each sensor node operates

Un-tethered and has a microprocessor and a small amount of memory for signal processing and task scheduling. Each node is equipped with one or more sensing devices such as acoustic microphone arrays, video or still cameras, infrared (IR), seismic, or magnetic sensors. Each sensor node communicates wirelessly with a few other local nodes within its radio communication range.

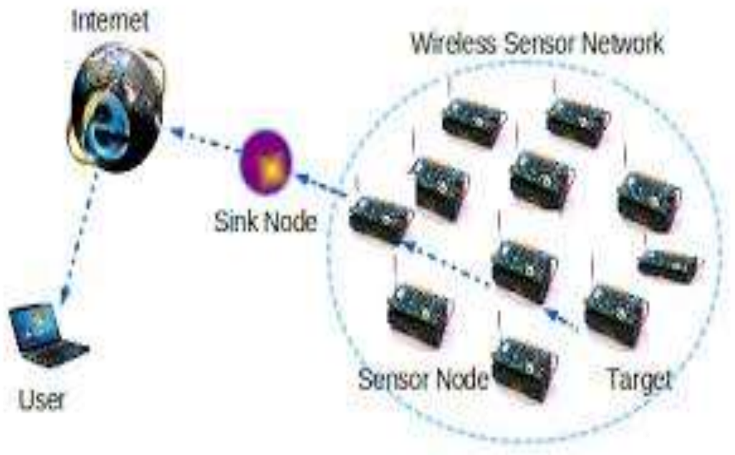

Fig 1.1 Sensor Network

Sensor networks extend the existing Internet deep into the physical environment. The resulting new network is orders of magnitude more expansive and dynamic than the current TCP/IP networks and is creating entirely new types of traffic that are quite different from what one finds on the Internet now. Information collected by and transmitted on a sensor network describes conditions of physical environments for example, temperature, humidity, or vibration and requires advanced query interfaces and search engines to effectively support user-level functions.

Sensor networks may inter-network with an IP core network via a number of gateways. A gateway routes user queries or commands to appropriate nodes in a sensor network. It also routes sensor data, at times aggregated and summarized, to users who have requested it or are expected to utilize the information. A data repository or storage service may be present at the gateway, in addition to data logging at each sensor. The repository may serve as an intermediary between users and sensors, providing a persistent data storage. It is well known that communicating 1 bit over the wireless medium at short ranges consumes far more energy than processing that bit

Wireless sensor networks are a trend of the past few years, and they involve deploying a large number of small nodes. The nodes then sense environmental changes and report them to other nodes over flexible network architecture. Sensor nodes are great for deployment in hostile environments or over large geographical areas. The sensor nodes leverage the strength of collaborative efforts to provide higher quality sensing in time and space as compared to traditional stationary sensors, which are deployed in the following two ways:

- Sensors can be positioned far from the actual phenomenon, i.e. something known by sense perception. In this approach, large sensors that use some complex techniques to distinguish the targets from environmental noise are required.
- Several sensors that perform only sensing can be deployed. The position of the sensors and communications topology is carefully engineered. They transmit time series of the sensed phenomenon to central nodes where computations are performed and data are fused.

1.2 Wireless Sensor Network vs. Ad hoc Network

A mobile ad hoc network (MANET), sometimes called a mobile mesh network, is a self configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. The difference between wireless sensor networks and ad-hoc networks are outlined below:

- The number of sensor nodes in a sensor network can be several orders of magnitude
- Higher than the nodes in an ad hoc network.
- Sensor nodes are densely deployed.
- Sensor nodes are prone to failures.
- The topology of a sensor network changes very frequently.
- Sensor nodes mainly use broadcast communication paradigm whereas most ad hoc
- Networks are based on point-to-point communication.
- ·Sensor nodes are limited in power, computational capacities, and memory.
- Sensor nodes may not have global identification (ID) because of the large amount of overheads and large number of sensors.
- ·Sensor networks are deployed with a specific sensing application in mind whereas ad-hoc networks are mostly Constructed For Communication Purpose.

1.3 Need For The System: Approximate Data Collection

Approximate data collection is a wise choice for long-term data collection in WSNs with constrained bandwidth. In many practical application scenarios with densely deployed sensor nodes, the gathered sensor data usually have inherent spatial-temporal correlations. For example, Fig. 1 shows the temperature readings of five nearby sensor nodes deployed in a garden more than 10 hours at night. The temperature readings recorded by the five nodes keep decreasing in the first 4 hours and then become stable in the next 6 hours, which exhibit apparent spatial and temporal correlations among themselves.

By exploring such correlations, the sensor data canbe collected in a compressive manner within prespecified, application-dependent error bounds. The data traffic can be reduced at the expense of data accuracy. The granularity provided by such approximate data collection is more than sufficient, especially considering the low measuring accuracy of sensors equipped on the sensor nodes. Study on approximate data collection is thus motivated by the need of long-term operation of large-scale WSNs, e.g., the GreenOrbs project.

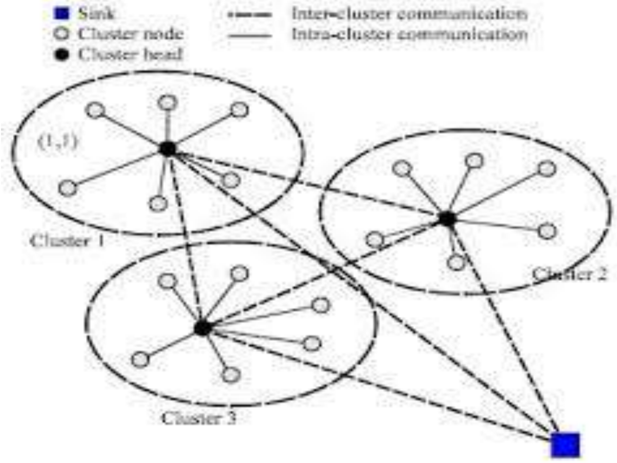

Fig 1.2 Data Aggregation

By exploring such correlations, the sensor data canbe collected in a compressive manner within prespecified, application-dependent error bounds. The data traffic can be reduced at the expense of data accuracy. The granularity provided by such approximate data collection is more than sufficient, especially considering the low measuring accuracy of sensors equipped on the sensor nodes. Study on approximate data collection is thus motivated by the need of long-term operation of large-scale WSNs, e.g., the GreenOrbs project.

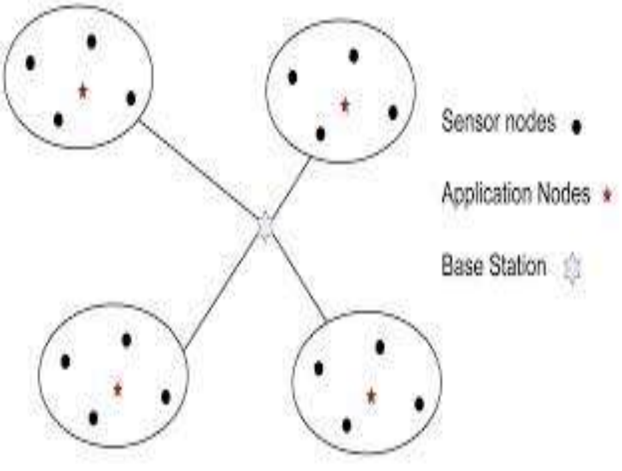

Fig 1.3 Cluster Formations

There are several factors to be considered in the design of an approach for approximate data collection. First, the data collection approach should be scalable. In many real applications, sensor networks consist of hundreds or even thousands of sensor nodes. For example, GreenOrbs has deployed 330 nodes and expects to deploy 1;000þ sensor nodes in a network.

In practice, in large WSNs, the information exchange between the sink and the related sensor nodes may consume considerable bandwidth and the acquisition of complete sensor data set of a WSN is too costly to be practical. Second, in approximate data collection, the spatial-temporal correlation model used for data suppression should be light-weight and efficient so as to meet the constraints on sensor node's memory and computation capacity. For densely deployed WSNs, many models can be used to describe temporal and/or spatial correlation of sensor data. But it is often nontrivial to build a light-weight correlation model to suppress spatial-temporal redundancy simultaneously.Most of the existing models are too expensive, i.e., consuming a large amount of computing capacity or storage capacity, to be run on the existing sensor nodes . Some of them are too simple to contain enough information and ignores the trend of sensor readings, or only consider either temporal correlation or spatial correlation separately. This thesis approach shows that simplicity and efficiency can be achieved by exploiting implicit sensor node cooperation and elaborately distributing data processing tasks to sensor nodes. Third, the data collection scheme should be self-adaptive to environmental changes. Note that physical environmental changes are usually complex and hard to be modeled comprehensively with a simple estimation model. For long-term data collection, the approximate data collection scheme should be capable of automatically adjusting its parameters according to the environmental changes so as to guarantee its correctness.

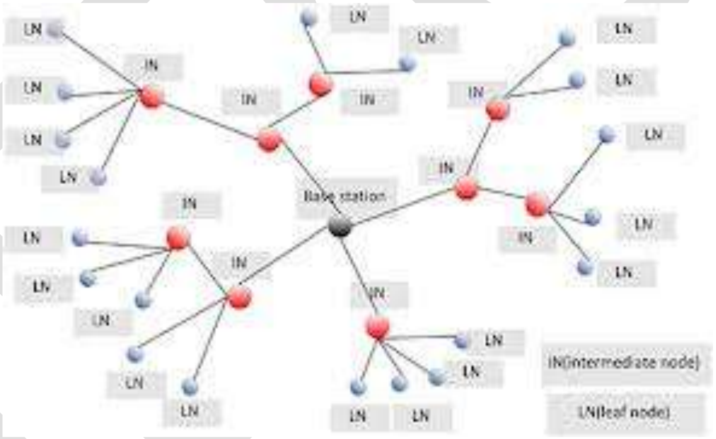

Fig 1.4 Data Collection

In this thesis, by leveraging the inherent spatial-temporal correlation in sensor data, an efficient approach is proposed for approximate data collection in WSNs to simultaneously achieve low communication cost and guaranteed data quality (namely bounded data errors). This thesis approach, Approximate Data Collection (ADC), is well designed to satisfy the above criterions. ADC achieves low communication cost by exploiting the fact that physical environments generally exhibit predictable stable state and strong temporal and spatial correlations, which can be used to infer the readings of sensors.Both the scalability and simplicity of ADC are achieved by exploiting implicit cooperation and distributing data processing among sensor nodes. ADC can discover local data correlations and suppress the spatial redundancy of sensor data in a distributive fashion. The distributed spatial data correlation discovery and spatial redundancy suppression is achieved by dividing a WSN into several clusters.

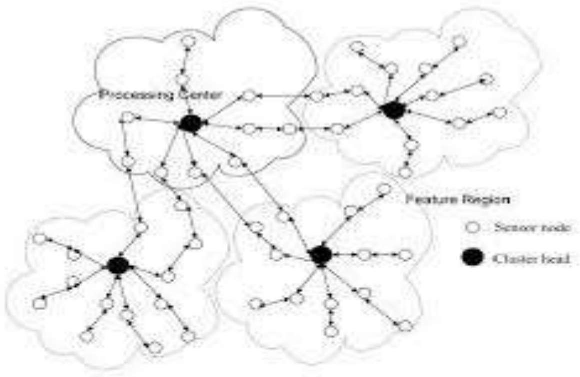

Fig 1.5 Cluster Members

The sink can estimate the sensor readings according to the model parameters updated by the cluster heads. This distributed data process scheme makes ADC can be easily applied to WSNs with different system scales. As the sensor network scale increases, ADC only needs to increase the number of clusters.

1.4 Scope Of Research Work

Furthermore, by using clustering-based distributed data process scheme, sensor data can be processed locally in ADC. First, each sensor node is responsible for processing sensor readings generated by itself. Second, the spatial redundancy of sensor data is suppressed by cluster heads that are close to the data source. There are no explicitly control data exchange between sensor nodes and their cluster heads. The sensor data process cost is distributed to all sensor nodes and the sensor data process burden of each cluster head can be easily controlled by adjusting the cluster size.

II. Problem Formulation

The problem of minimizing the schedule length for raw-data convergecast on single channel is more. Convergecast, namely the collection of data from a set of sensors toward a common sink over a treebased routing topology, is a fundamental operation in wireless sensor networks (WSN). In many applications, it is crucial to provide a guarantee on the delivery time as well as increase the rate of such data collection. For instance, in safety and mission-critical applications where sensor nodes are deployed to detect oil/gas leak or structural damage, the actuators and controllers need to receive data from all the sensors within a specific deadline, failure of which might lead to unpredictable and catastrophic events. This falls under the category of one-shot data collection. On the other hand, applications such as permafrost monitoring require periodic and fast data delivery over long periods of time, which falls under the category of continuous data collection.

For periodic traffic, it is well known that contention free medium access control (MAC) protocols such as TDMA (Time Division Multiple Access) are better fit for fast data collection, since they can eliminate collisions and retransmissions and provide guarantee on the completion time as opposed to contention-based protocols. However, the problem of constructing conflictfree (interference-free) TDMA schedules even under the simple graph-based interference model has been proved to be NP-complete. In this project, consider a TDMA framework and design polynomial-time heuristics to minimize the schedule length for both types of convergecast.

It also find lower bounds on the achievable schedule lengths and compare the performance of our heuristics with these bounds. The problem of joint scheduling and transmission power control for constant and uniform traffic demands. It can be overcome by Aggregate converge cast and One-Shot Raw-Data converge cast algorithms.

III. Objectives Of The Research

The research work main objective is information be collected from a wireless sensor network organized as tree. To address this, a number of different techniques using realistic simulation models under the many-to-one communication paradigm known as convergecast are evaluated. Time scheduling on a single frequency channel with the aim of minimizing the number of time slots required (schedule length) to complete a convergecast is considered. Next, scheduling with transmission power control is combined to mitigate the effects of interference, and show that while power control helps in reducing the schedule length under a single frequency, scheduling transmissions using multiple frequencies is more efficient.

It investigated the impact of transmission power control and multiple frequency channels on the schedule length, where the proposed constant factor and logarithmic approximation algorithms on geometric networks (disk graphs). Raw-data convergecast has been studied in a distributed time slot assignment scheme is proposed in the project to minimize the TDMA schedule length for a single channel. The project also compares the efficiency of different channel assignment methods and interference models, and proposes schemes for constructing specific routing tree topologies that enhance the data collection rate for both aggregated and raw-data convergecast.

IV. Related work

Adaptive Approximate Data Collection

Since sensor readings change slowly according to the change of physical phenomena, the adaptive data approximation algorithm should be self-adaptive to the changes of the sensor readings timely. The proposed data approximation algorithm consists of two parts: data approximation learning algorithm and data approximation monitoring (for cluster heads and sink node) algorithm.

The data approximation learning algorithm runs on every cluster head and is responsible for finding a Δ-loss approximation of the true sensor data of each cluster. The data approximation monitoring algorithm consists of two parts. One runs on every cluster head continuously. It monitors the changes of the parameters of the local estimation and decides whether to send an update message to the sink node or not. The other part,which runs on the sink node, is responsible for updating the Δ-loss approximation according to the update messages from each cluster head.

(i).Routing Tree

The sink node is treated as root node. All other nodes neighbor to sink behaves as intermediate nodes. The nodes responsible for collecting the data behave as leaf nodes. The intermediate nodes aggregates the data received from leaf nodes are sends to sink node.

(ii). The Data Approximation Learning Algorithm

The data approximation learning algorithm guarantees that the predictor set SS stored in the sink node is a Δ-loss approximation of IF at all times. Each cluster head starts the data approximation monitoring algorithm after the data approximation learning algorithm. The data approximation monitoring algorithm updates all local estimation data according to the received local estimation update messages and checks the estimation error of each Θ-similarity set every T seconds. Each sensor node requires WS*T seconds to check the correctness of it local estimation model, the estimation error check is delayed by WS*T seconds. If the radius of any Θ-similar set exceeds Δ, the cluster head will adjust its local Θ-similarity sets and send the changes to the sink node. The sink node updates SS according to the update messages from the cluster heads.

The Data Approximation Learning Algorithm

```
Generate correlation graph Gi(V, E, t)
i = 0;
while |V| > 0 do
    v = FindLargestOutDegree(V);
    w[i].representation_node=v;
    w[i].similarity_set=AllNeighbor(v);
    V - = {v};
    V - =  w[i].similarity_set;
    i++;
end while
return w;
```

(iii). Monitoring Algorithm for the Cluster Heads

The details of the data approximation monitoring algorithm for cluster heads are shown in Algorithm. The algorithm first updates all local estimations according to all local estimation update messages M received in last T seconds (line 1). Line 2-12 search each Θ-similar set and find out all sensor nodes that are no longer Θ-similar to their representation nodes, then add them into node list C C. All empty Θ-similar sets are removed (line 9-10).

Each sensor node in CC tries to find a Θ-similar set to join in by invoking the procedure Join() (line 14). If there is no such a set, a new Θ-similar set will be created for this node by invoking the procedure CreatNewSet() (line 16). Line 20 sends the update messages to the sink node.

The data approximation monitoring algorithm only requires two kinds of update messages: the Θ-similar set creating message and the Θ-similar set updating message. The former creates a new Θ-similar set at the sink node, while the latter is used to update the predictor of a Θ-similar set or add new sensor nodes into it. Note that explicitly sending a message for removing a sensor node from a Θ-similar set is not necessary, because no sensor node belongs to two or more Θ-similar sets simultaneously. Adding a node into a Θ-similar set means removing it from another one.

Monitoring Algorithm for the Cluster Heads

```
1: UpdateMessagePrc(M);
2: for all $W \in G$ do
    3: for all $s \in W$ do
        4: if $D(s,W, t) > \Delta$ then
            5: $CC = CC \cup \{s\}$;
            6: $W-=\{s\}$;
        7: end if
    8: end for
    9: if $W = \emptyset$ then
        10: $G-=W$;
    11: end if
12: end for
13: for all $s \in CC$ do
    14: flag=Join(s);
```

```
15: if flag==0 then
16: W = CreatNewSet(s);
17: $G = G \cup W$;
18: end if
19: end for
20: SendUpdatemsg();
```

(iv). Monitoring Algorithm For The Sink Node

The details of the data approximation monitoring algorithm for the sink node are shown in Algorithm. After receiving an updating message M, the sink node first checks its message type. If it is a Θ-similar set creating message, it first removes all the nodes contained in M from current exiting Θ-similar sets (line 2), then creates a new Θ-similar set and adds all these nodes contained in M into the new Θ-similar set (line 3). If M is a Θ-similar set updating message, the sink node first removes all the nodes contained in M from current exiting Θ-similar sets (line 7), then updates the predictor of the specified Θ-similar set or add all the node contained in M into the specified Θ-similar set (line 8). Finally, all empty sets are removed (line 10-14).

Monitoring Algorithm for the Sink Node

```
1: if msgtype is Θ-similar set creating message then
2: Remove(M);
3: W = CreatNewSet(M);
4: $G = G \cup \{W\}$;
5: end if
6: if msgtype is Θ-similar set updating message then
7: Remove(M);
8: SetUpdate(M);
9: end if
10: for all $W \in G$ do
11: if $W = \emptyset$ then
```

```
    12: G-= W;

    13: end if

    14: end for
```

(v).Transmission Plan

For each query Qi with a routing tree Ti, during each period, first each leaf node in Ti adds the source data to its transmission plan. Then, every internal node in (noted as a relay node for query Qi) only generates one unit of data by aggregating all received data with its own data (if it has), while it may receive multiple data units from its children.

(vi).Packet Scheduling

Packet scheduling at each node that contains data units in its transmission plan is occurred. The nodes are divided into two complementary groups: leaf nodes and intermediate nodes. It ensures that all leaf nodes transmit at even time-slots only, and all intermediate nodes transmit at odd time-slots only.

(vii).Aggregation Degree Setting

Aggregation degree (number of packets that can be aggregated (data can be summed, maximum data, minimum data or average data) is set at each node.

(viii).Aggregation and Transmission Based On Degree

Data is aggregated based on degree and transmission occurred according to degree.

Experimental and Results

The following Table2.1 describes experimental result for proposed system performance rate analysis. The table Contains number of cluster, cluster size and number of aggregated data and average aggregated data details are shown

S.No	Number Of Cluster	Cluster A	Cluster B	Cluster C	Cluster D	Cluster E	Cluster F
1	2 Cluster	56	89	65	67	67	67
2	3 Cluster	67	89	78	89	89	89
3	4 Cluster	78	68	89	56	56	89

4	5 Cluster	89	56	67	67	68	67
5	6 Cluster	56	67	56	56	67	56
6	7 Cluster	67	72	67	67	67	58
7	8 Cluster	78	89	89	89	89	67
8	9 Cluster	89	67	67	66	76	76
	No. Of Aggregated Data	580	597	578	557	579	569
	Avg %	72.5	74.625	72.25	69.625	72.375	71.125

Table2.1 Cluster Size: Proposed System and Performance Rate

The following Table 2.2 describes experimental result for existing system over all experimental result analysis. The table contains aggregated cluster, number of aggregated data cluster data and average aggregated data details are shown

Aggregated Cluster	No. Of. Aggregated Data	Avg % Aggregated
Cluster A	580	72.5
Cluster B	597	74.62
Cluster C	578	72.25
Cluster D	557	69.62
Cluster E	579	72.37
Cluster F	569	71.12

Table 2.2 Overall Experimental Results - Proposed System

The following Fig **2.1**describes experimental result for proposed data transfers for hybrid method rate analyses are shown.

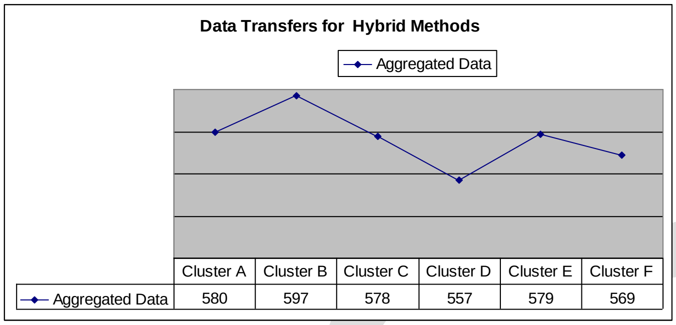

Fig 2.1 Proposed System - Aggregated Data

The following Fig **6.4** describes experimental result for proposed system aggregation scheme analyses are shown

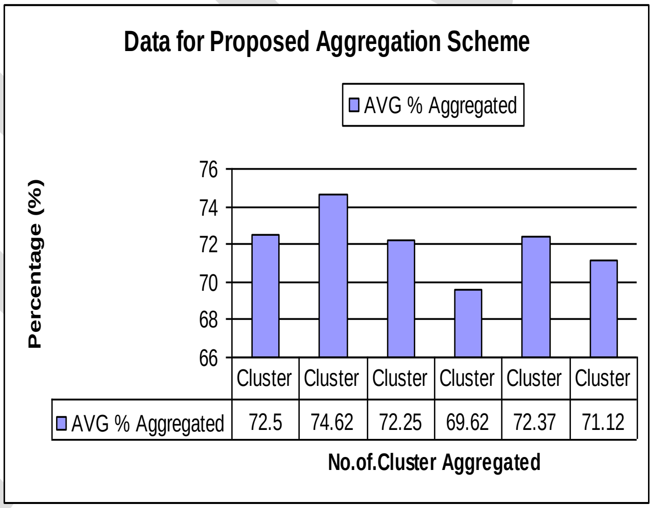

Fig 6.3 Aggregation Scheme- Proposed System

CONCLUSION

In research work are implementations by approximate data collection between wireless sensor networks. In this proposed system exploring application level data collection process. The wireless sensor network is collecting data between one network to another ad hoc networks data is stable and strong temporal and spatial correlation between sensor readings. Our work detects data similarities among the sensor nodes by comparing their local estimation models rather than their original data. The simulation results show that this approach can greatly reduce the amount of messages in wireless communications.

In this research work fast convergecast in WSN node communication using a TDMA protocol to minimize the schedule length is considered. In this thesis work additionally data collection between tree based and schedule process. The degree of node level is finding parent and child node level, the parent node is send data into server node and child node into another sink node details. Therefore, time complexity minimum for data collection between sensor node details. The system addressed the fundamental limitations due to interference and half-duplex transceivers on the nodes and explored techniques to overcome the same. It is found that while transmission power control helps in reducing the schedule length, multiple channels are more effective

REFERENCES

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S.Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, “A Macroscope in the Red Woods,” Proc. Third Int’l Conf. Embedded Networked Sensor Systems (SenSys ’05), 2005.

[2] M. Li, Y. Liu, and L. Chen, “Non-Threshold Based Event Detection for 3D Environment Monitoring in Sensor Networks,” IEEE Trans.Knowledge and Data Eng., vol. 20, no. 12, pp. 1699-1711, Dec. 2008.

[3] Z. Yang and Y. Liu, “Quality of Trilateration: Confidence based Iterative Localization,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 5, pp. 631-640, May 2010.

[4]G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and Yield in a Volcano Monitoring Sensor Network,” Proc. Seventh Symp. Operating Systems Design and Implementation (OSDI ’06), 2006.

[5] R. Cardell-Oliver. Rope: A reactive, opportunistic protocol for environment monitoring sensor network. In Proc. The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II), May 2005.

[6] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, “Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest,” Proc. Sevent h ACM Conf. Embedded Networked Sensor Systems (SenSys ’09), 2009.

[7] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In Proc. the Workshop on Data Communications in Latin America and the Caribbean, Apr. 2001.

[8] R. E. Grumbine, "What Is Ecosystem Management?" Conservation Biology, vol. 8, pp. 27-38, 1994.

[9] F. L. Bunnell and D. J. Vales., "Comparison of Methods for Estimating Forest Overstory Cover: Differences among Techniques," Can. J. Forest Res., vol. 20, 1990.

[10] D. Chu, A. Deshpande, J.M. Hellerstein, and W. Hong, "Approximate Data Collection in Sensor Networks Using Probabilistic Models," Proc. 22nd Int'l Conf. Data Eng. (ICDE '06), 2006.

[11] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-Driven Data Acquisition in Sensor Networks," Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04), 2004.

[12] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In ACM SIGMOD, 2003.

[13] C. Wan, S.B. Eisenman, and A.T. Campbell, "Coda: Congestion Detection and Avoidance in Sensor Networks," Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03), 2003.

[14] A. Silberstein, R. Braynard, and J. Yang, "Constraint Chaining: On Energy-Effcient Continuous Monitoring in Sensor Networks," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), 2006.

[15] I. Solis and K. Obraczka. Efficient continuous mapping in sensor networks using isolines. In Proc. of the 2005 Mobiquitous, July 2005.

[16] X. Meng, L. Li, T. Nandagopal, and S. Lu. Event contour: An efcient and robust mechanism for tasks in sensor networks. Technical report, UCLA, 2004.

[7] C. Olston and J. Widom, "Best Effort Cache Sychronization with Source Cooperation," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '02), 2002