Mobile Info Gathering with Load Balanced Bundling and Dual Data Uploading in Wireless Sensor Networks

Mr. Rahul Shinde[1]	Mr. Pankaj Kanade[2]	Ms. Prajakta Dhondge[3]	Ms. Amruta Kulkarni[4]	Prof.H.D. Sonawane[5]
BE Computer	BE Computer	BE Computer	BE Computer	M.Tech Software
Dept. of Computer	Dept. of Computer	Dept. of Computer	Dept. of Computer	Dept. of Computer
BVCOERI, Nashik	BVCOERI, Nashik	BVCOERI, Nashik	BVCOERI, Nashik	BVCOERI, Nashik

rbshindepatil@gmail.com[1], pankaj.kanade1@gmail.com[2], bhushan.patil2201@gmail.com[3], amukulkarni171990@gmail.com[4], hd.sonawane@gmail.com[5]

Abstract

This is referred to as the basics earalainsa DDU, which load balancing cluster and dual upload information, distributed jobs. The objective is the conclusion drawn stayed good scalability, long lifetime and low network data collection. The clusters themselves sensor layer, a distributed load balancing bundle (LBB) to organize self-proposed algorithms sensors. In contrast to existing methods of the bundle, construction and our plan to balance the load dual data upload feature to work more than one cluster head in each cluster. At the head of the bundle layer is chosen among the guarantee inter-connectivity bundle bundle carefully spread range. Mutual cooperation of several major energy savings within a bundle bundle to launch inter-bundle communication. Inter-bundle is moved forward planning SenCar trajectory information collected through the projection head. At the mobile collector layer, SenCar two antennas, at the same time, multi-user bundle is equipped with two heads to upload data each time SenCar using multiple input and multiple output (myu MIMO) system. SenCar way planning is fully optimized use of double data upload capability for selecting each group to organize voting points. Selected by visiting each polling point, SenCar can efficiently collect data bundle and the head steady traffic data sink data. The proposed comprehensive simulations are conducted to evaluate the effectiveness of the plan earalainsa DDU. The result is two bundles on the head of each bundle to a maximum, then the node that received earalainsa DDU and collect 60 percent of the information through the energy savings of around 50 percent energy savings static data multi- hop relay comparison show cast cluster head, and 20 per cent of the collection of traditional mobile data small data collection time.

Keywords
WSN, load balanced bundling, MU-MIMO, mobility control, polling point.

1. INTRODUCTION

Mobile computing is the discipline for creating an information management platform, which is free from spatial and temporal constraints. The freedom from these constraints allows its users to approach and process desired information from anywhere in the space. The state of the user, static or mobile, does not affect the information management capability of the mobile platform. A user can continue to approach and manipulate desired data while traveling on plane, in car, on ship, etc. Thus, the discipline creates an illusion that the desired data and satisfactory processing power are available on the spot, where as in reality they may be located far away. Otherwise Mobile computing is a generic term used to refer to a variety of devices that allow people to access data and information from where ever they are..

ISSN : 2394-0905

Figure1. Structure of mobile computing

Different types of devices used for the mobile computing:

1. Personal digital assistant/enterprise digital assistant
2. Smartphones
3. Tablet computers
4. Netbooks
5. Ultra-mobile PCs
6. Wearable computers
7. Palmtops/pocket computers

2. LITERATURE SURVEY

We consider the problem of collecting a large amount of info from several different hosts to a single destination in a wide-area network. This problem is important since improvements in info collection times in many applications such as wide-area upload applications, high-performance computing applications, and data mining applications are crucial to performance of those applications. Often, due to bottleneck conditions, the paths chosen by the network may have poor throughput. By choosing an alternate route at the application level, we may be able to obtain substantially faster completion time. This info collection problem is a nontrivial one because the issue is not only to avoid congested link(s), but to devise a coordinated transfer schedule which would afford maximum possible usage of available network resources. Our approach for computing coordinated info collection schedules makes no assumptions about knowledge of the topology of the network or the amplitude available on individual links of the network. This approach provides significant performance improvements under various degrees and types of network congestion. To show this, we give a complete comparison study of the various approaches to the data collection problem which considers performance, robustness, and adaptation characteristics of the different info collection methods. The adaptation to network conditions characteristics are important as the above applications are long lasting, i.e., it is likely changes in network conditions will occur during the info transfer process. In general, our approach can be used for solving capricious info movement problems over the Internet. We use the Bistro platform to illustrate one application of our techniques.

In a heterogeneous wireless sensor network (WSN), relay nodes (RNs) are adopted to relay info packets from sensor nodes (SNs) to the base station (BS). The distribution of the RNs can have a significant impact on connectivity and lifetime of a WSN system. This paper studies the effects of random distribution strategies. We first discuss the biased energy consumption rate problem associated with uniform random distribution. This problem leads to insufficient energy utilization and shortened network lifetime. To overcome this problem, we propose two new random distribution strategies, namely, the lifetime-oriented deployment and hybrid distribution. The former solely aims at balancing the energy consumption rates of RNs across the network, thus extending the system lifetime. However, this distribution scheme may not provide sufficient connectivity to SNs when the given number of RNs is relatively small. The latter reconciles the concerns of affinity and lifetime extension. Both single-hop and multihop communication models are considered in this paper. With a combination of theoretical analysis and simulated assessment, this study explores the trade-off between connectivity and lifetime extension in the problem of RN deployment. It also provides a guideline for efficient deployment of RNs in a large-scale heterogeneous WSN.

Most sensor networks are used to collect information from the physical world. Examples include sensor networks deployed to monitor micro-climates in agriculture farms and distributions that measure energy consumption in office or residential buildings. The nodes in these networks collect information about the physical world using their sensors and relay the sensor readings to a central base station or server utilizing multi-hop wireless communication.

Collecting information reliably and efficiently from the nodes in a sensor network is a challenging problem, specially due to the wireless dynamics. Multihop routing in a dynamic wireless environment requires that a protocol can adapt quickly to the changes in the network (agility) while the energy-constrains of sensor networks precept that such mechanisms not require too much communication among the nodes (efficiency). CTP is a collection routing protocol, that achieves both agility and efficiency, while offering highly reliable info delivery in sensor networks.

CTP has been used in research, teaching, and in commercial products. An experience with CTP has also update the design of the IPv6 Routing Protocol for Low power and Lossy Networks (RPL).

Most current geographic routing protocols on sensor networks concentrates on finding ways to guarantee data forwarding from the source to the destination, and not many protocols have been done on accumulation and aggregating data of sources in a local and adjacent region. However, data generated from the sources in the region are often redundant and highly correlated. Accordingly, accumulation and aggregating data from the region in the sensor networks is important and necessary to save the energy and wireless resources of sensor nodes. We introduce the concept of a local sink to address this issue in geographic routing. The local capsize is a sensor node in the region, in which the sensor node is temporarily selected by a global capsize for accumulation and aggregating data from sources in the region and convey the aggregated data to the global capsize. We next design a Single Local Sink Model for determining optimal location of single local capsize. Because the buffer size of a local capsize is limited and the deadline of data is constrained, single local capsize is capable of carrying out many sources in a large-scale local and adjacent region. Hence, we also extend the Single Local capsize Model to a Multiple Local Sinks Model. We next propose a data gathering mechanism that gathers info in the region through the local capsize and delivers the aggregated data to the global capsize. Simulation results show that the proposed mechanism is more efficient in terms of the energy consumption, the data delivery ratio, and the deadline miss ratio than the existing mechanisms.
Energy efficiency is critical for wireless sensor networks. The data-gathering process must be carefully designed to conserve energy and boost network lifetime. For applications where each sensor continuously monitors the environment and periodically reports to a base station, a tree-based topology is often used to collect info from sensor nodes. In this work, we first study the construction of a data-gathering tree when there is a single base station in the network. The objective is to maximize the network lifetime, which is defined as the time until the first node bankrupts its energy. The problem is shown to be NP-complete. We design an algorithm that starts from an arbitrary tree and constant reduces the load on bottleneck nodes (nodes likely to soon deplete their energy due to high degree or low remaining energy). We then extend our work to the case when there are multiple base stations and study the composition of a maximum-lifetime data-gathering forest. We show that both the tree and forest construction algorithms terminate in polynomial time and are provably near optimal. We then verify the adequacy of our algorithms via numerical comparisons.

3. CONCEPT

In the computation phase, each sensor acquaints itself with all the neighbors in its proximity. If a sensor is an isolated node (i.e., no neighbor exists), it claims itself to be a cluster head and the cluster only contains itself. differently, a sensor, say, si, first sets its status as "tentative" and its initial priority by the percentage of residual energy. Then, si sorts its neighbors by their initial preferences and picks neighbors with the highest initial priorities, which are temporarily treated as its candidate peers. We denote the set of all the candidate peers of a sensor by A. It implies that once si successfully allegations to be a cluster head, its up-to-date candidate peers would also automatically become the cluster heads, and all of them form the CHG of their cluster. si sets its priority by counting up its initial priority with those of its candidate peers. In this way, a sensor can choose its favorable peers along with its status decision.

3.1 Status Claim

In the second module, each sensor determines its status by iteratively updating its local information, refraining from promptly claiming to be a cluster head. We use the node degree to rule the maximum number of iterations for each sensor. Whether a sensor can finally become a cluster head primarily depends on its priority. Specifically, we partition the priority into three zones by two thresholds, th and tm (th > tm) , which enable a sensor to declare itself to be a cluster head or member, respectively, before reaching its maximum number of iterations. All the while the iterations, in some cases, if the priority of a sensor is greater than th or less than tm compared with its neighbors, it can immediately decide its final status and quit from the iteration. We denote the probable cluster heads in the neighborhood of a sensor by a set B. In each iteration, a sensor, say, si, first tries to probabilistically include itself into si:B as a experimental cluster head if it is not in already. Once successful, a packet includes its node ID and priority will be sent out and the sensors in the closeness will add si as their potential cluster heads upon receiving the packet. Then, si checks its current potential cluster heads. If they do exist, there are two cases for si to make the final status decision; any other way, si would stay in the experimental status for the next round of iteration.

3.2 Cluster Forming

The third module is cluster forming that decides which cluster head a sensor should be correlated with. The criteria can be described as follows: for a sensor with experimental status or being a cluster member, it would anyway affiliate itself with a cluster head among its candidate peers for load balance ambition. In the rare case that there is no cluster head among the candidate peers of a sensor with experimental status, the sensor would claim itself and its current candidate peers as the cluster heads.

3.3 Synchronization among Cluster Heads

To perform data collection by TDMA techniques, intracluster time synchronization among entrenched cluster heads should be premeditated. The fourth phase is to synchronize local clocks among cluster heads in a CHG by beacon messages. First, each cluster head will deliver out a beacon message with its initial arrangement and local clock info to other nodes in the CHG. Then it examines the received beacon messages to see if the priority of a beacon message is more advanced. If yes, it adjusts its local clock granting to the timestamp of the beacon message. In our framework, such synchronization among cluster heads is only act while SenCar is collecting data. Because data collection is not very constant in most mobile data gathering applications, message overhead is certainly manageable within a cluster.

4. SYSTEM REQUIREMENT

Hardware Requirements:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

Software Requirements:

- Operating system : - Windows XP.
- Coding Language : C#.NET
- Data Base : MS SQL SERVER 2005

5. WORKING

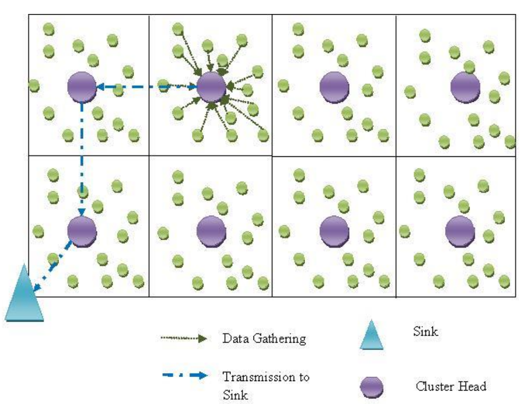

Figure2. System Architecture

5.1 Input Design

The input design is the link between the info system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction info in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the info directly into the system. The design of input target on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is arranged in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the info should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

5.2 Output Design

A quality output is one, which meets the fulfillment of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is dogged how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Capable and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an standardized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When study design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that have information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal crucial events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

5.3. Data Flow Diagram:

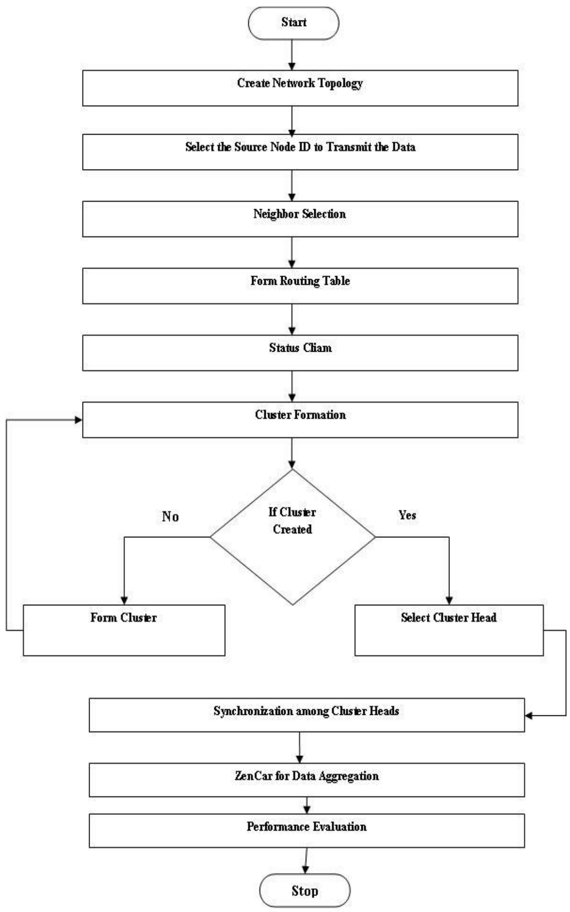

Figure3. Data Flow Diagram

5.4 Objectives

1. Input Design is the process of adapt a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the appropriate direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle big volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data operates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are arranged as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

6. ADVANTAGES OF PROPOSED SYSTEM:

- In divergence to clustering techniques proposed in previous works, our algorithm balances the load of intra-cluster aggregation and enables dual data uploading between multiple cluster heads and the mobile collector.

- Different against other hierarchical schemes, in our algorithm, cluster heads do not relay data packets from other clusters, which effectively allay the burden of each cluster head. Instead, forwarding paths among clusters are only used to route small-sized identification (ID) information of cluster heads to the mobile collector for optimizing the data collection circuit.

- Our work mainly distinguishes from other mobile collection schemes in the utilization of MUMIMO technique, which enables dual data uploading to shorten data transmission latency. We coordinate the mobility of Sensor to fully enjoy the benefits of dual data uploading, which ultimately leads to a data collection tour with both short moving trajectory and short data uploading time.

7. DISADVANTAGES OF EXISTING SYSTEM:

- In relay routing schemes, minimizing energy consumption on the forwarding path does not necessarily perpetuate network lifetime, since some critical sensors on the path may run out of energy faster than others.

- In cluster-based schemes, cluster heads will inevitably consume much higher energy than other sensors due to handling intra-cluster aggregation and inter-cluster data forwarding.

- Though using mobile collectors may alleviate non-uniform energy consumption, it may result in unsatisfactory data collection latency.

8. CONCLUSION

We have suggested this earalainsa DDU mobile data collection in the framework of a WSN. The sensor substrate, the substrate and the sensor layer consists of a group head. The sensor self-organization among the load balanced cluster distribution of energy-efficient launch inter-cluster communication partner accepts employment CHGs, faster data collection uses double data upload, and enjoy the benefits of full mobility of emayu MIMO sensor. Study shows effectiveness of the proposed framework of our performance. The results show that the earalainsa DDU greatly alleviating burdens nodes on the way, and SISO mobile to collect information and cluster head 60 per cent energy savings compared to the cluster, which received less than 20 per cent of the data collection time in the head, among cut energy consumptions by working balance. We have explored a number of good results in different energy windows and overhead cluster head. Finally, we want to mark out that there are some interesting problems that can be studied in our future work. The first problem is to find out how to vote and the right combinations of points in each group. A discretionary allocation plan must be developed to find the optimal polling station constantly point for each group. And it was a similar problem in order to achieve optimal overall aerial diversity for compatible pairs. Another problem is how to schedule the upload MIMO from many clusters. An algorithm should be calculated based on the future of this MIMO transmission scheduling algorithm adapts.

9. ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on 'Mobile Info Gathering with Load Balanced Clustering and Dual Data Uploading in Wireless Sensor Networks.

I would like to take this opportunity to thank my internal guide Prof. H. D. Sonawane sir for giving me all the help and guidance I needed. I am really thankful to them for their kind support. Their valuable suggestions were very helpful.

I am also grateful to Prof. H. D. Sonawane sir, Head of Computer Engineering Department, BVCOE&RI for his indispensable support, suggestions.

Last but not least our special thanks to our staff member, lab technicians for providing miscellaneous, resources such as laboratory with all needed software platforms, continuous Internet connection for Our Project.

10. REFERENCES

[1] B. Krishnamachari, Networking Wireless Sensors. Cambridge, U.K.: Cambridge Univ. Press, Dec. 2005.

[2] R. Shorey, A. Ananda, M. C. Chan, and W. T. Ooi, Mobile, Wireless, Sensor Networks. Piscataway, NJ, USA: IEEE Press, Mar. 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol. 40, no. 8, pp. 102–114, Aug. 2002.

[4] W. C. Cheng, C. Chou, L. Golubchik, S. Khuller, and Y. C. Wan, "A coordinated data collection approach: Design, evaluation, and comparison," IEEE J. Sel. Areas Commun., vol. 22, no. 10, pp. 2004–2018, Dec. 2004.

[5] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," IEEE Trans. Mobile Comput., vol. 9, no. 2, pp. 145–159, Feb. 2010.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proc. 7th ACM Conf. Embedded Netw. Sensor Syst., 2009, pp. 1–14.

[7] E. Lee, S. Park, F. Yu, and S.-H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," IEEE Trans. Consum. Electron., vol. 56, no. 3, pp. 1433–1441, Aug. 2010.

[8] Y. Wu, Z. Mao, S. Fahmy, and N. Shroff, "Constructing maximum- lifetime data-gathering forests in sensor networks," IEEE/ ACM Trans. Netw., vol. 18, no. 5, pp. 1571–1584, Oct. 2010.

[9] X. Tang and J. Xu, "Adaptive data collection strategies for lifetime- constrained wireless sensor networks," IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 6, pp. 721–7314, Jun. 2008.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Trans. Wireless Commun., vol. 1, no. 4, pp. 660–660, Oct. 2002.

[11] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in IEEE Conf. Comput. Commun., pp. 366–379, 2004.

[12] D. Gong, Y. Yang, and Z. Pan, "Energy-efficient clustering in lossy wireless sensor networks," J. Parallel Distrib. Comput., vol. 73, no. 9, pp. 1323–1336, Sep. 2013.

[13] A. Amis, R. Prakash, D. Huynh, and T. Vuong, "Max-min d-cluster formation in wireless ad hoc networks," in Proc. IEEE Conf. Comput. Commun., Mar. 2000, pp. 32–41.

[14] A. Manjeshwar and D. P. Agrawal, "Teen: A routing protocol for enhanced efficiency in wireless sensor networks," in Proc. 15th Int. IEEE Parallel Distrib. Process. Symp., Apr. 2001, pp. 2009–2015.