Improvement of limited Storage Placement in Wireless Sensor Network

[1] Prof. Dr.S.S.Apte, [2] Sagar M Mane

[1](Head, Computer Science & Engg, W.I.T. Solapur/ Solapur University, India)
[2](Computer Science & Engg, W.I.T. Solapur/ Solapur University, India)

Abstract: *In sensor network a large amount of data need to be collected for future information retrieval. The data centric storage has become an important issue in sensor network. Storage nodes are used in this paper to store and process the collected data. This paper considers the storage node placement problem aiming to place limited storage nodes in sensor network to minimize the total energy cost for collecting the raw data and replying queries at the storage nodes. In this paper a strong data access model for placing storage nodes in sensor network is presented. We consider an application in which sensor networks provide real time data services to user. The main aim of this paper is to reduce the cost for raw data transfer, query diffusion, query reply by defining the best location of storage nodes in sensor network.*
Keywords – *Wireless sensor network, data query, data storage, data reply*

I. Introduction

One of the key challenges in wireless sensor network is the storage and querying of useful sensor data. The wireless sensor network is built of nodes, where each node is connected to one or more sensors. Sensor networks deployed for different computing applications,.e.g.,sensing environmental or earth condition and monitoring people's behaviors, generates a large amount of data over a long period of time. Storage is an essential factor of any data centric sensor network application. One of the main challenges in these applications is how to search and store the collected data. The collected data can either be stored in the network sensors, or transmitted back to the sink and stored there for future retrieval. This design is ideal since data are stored in a central place for permanent access. Placing limited storage nodes is related to the sensor network. Query is the most important part of sensor network since in aspect sensor network provides the information about environmental condition to the end user. Therefore, we aim to minimize the total energy cost and data query by accurately deploying the storage nodes in sensor network. In section III we discuss the Problem Formulation. In section IV, we discuss the algorithm for placing limited storage nodes in wireless sensor network. In section V, we present the conclusion and future work.

II. Related Work

There has been a lot of prior research work on data querying models in sensor network. In early models [1, 2, 3], query is spread to every sensor node by flooding messages. Sensors nodes send data back to the sink in the reverse direction of query messages. Those methods do not consider the storage concern in sensor networks. PRESTO [4] is a recent research works on storage architecture for wireless sensor networks. A proxy layer is introduced between sensor nodes and user terminals and proxy nodes can cache previous query responses. When a query arrives in a proxy node, it first checks if the cached data can satisfy the query before forwarding the query to other nodes. Compared with the storage nodes in this paper, Nodes in PRESTO have no resource constraints in term of computation, power, storage and communication. It is a more familiar storage architecture that does not take the characteristics of data generation or query into consideration.
Data-centric storage schemes [5, 6, 7] store data to different places in sensor networks according to different data types.
In [6, 7], the authors propose a data-centric storage scheme for sensor networks, which inherits ideas from distributed hash table. The home site of a data is obtained by applying a hash function on the data type.
LEACH [8] is a clustering based routing protocol, in which cluster heads can fuse the data collected from its neighbors to reduce communication cost to the sink. LEACH has a similar structure to our scheme, having cluster heads aggregate and forward data to the sink. However, LEACH aims to reduce data transmission by aggregating data; it does not address storage problem in sensor networks

III. Problem Formulation

In this paper, we consider an application in which sensor networks provide real-time data services to users. A sensor network is given with one defined sensor identified as the sink (or base station), access point and many normal sensors, each of which generates (or collects) data from its environment. Users or application program specify the data they need by submitting queries to the sink and they are usually interested in the latest readings generated by the sensors. To reply to queries, one typical solution, shown in fig.1, is the sinks have all the data.

This requires each sensor to send its readings back to the access point immediately every time it produces new data. Transferring all raw data could be very expensive and is not always required. Alternatively, we allow sensors node to hold their raw data and to be aware of the different queries, then raw data can be managed to contain only the readings that users are interested in and the reduced reply size, instead of the whole raw data readings, can be send back to the sink. This design is illustrated in Fig. 1, where the black sensor nodes, called storage nodes, are allowed to hold raw data. The base station diffuses queries to the access point by broadcasting to the sensor network and then access point broadcast the queries to storage sensors and these storage sensors reply to the queries by sending the processed data back to the storage node. Compared to the earlier solution, this approach reduces cost of the raw data transfer because some raw data transmissions are replaced by query reply. On the other hand, this scheme incurs an extra query diffusion cost (as figured by the dashed arrows). In this paper, we are interested in vital designing a data access model to minimize energy cost associated with query diffusion, raw data transfers, and query replies.
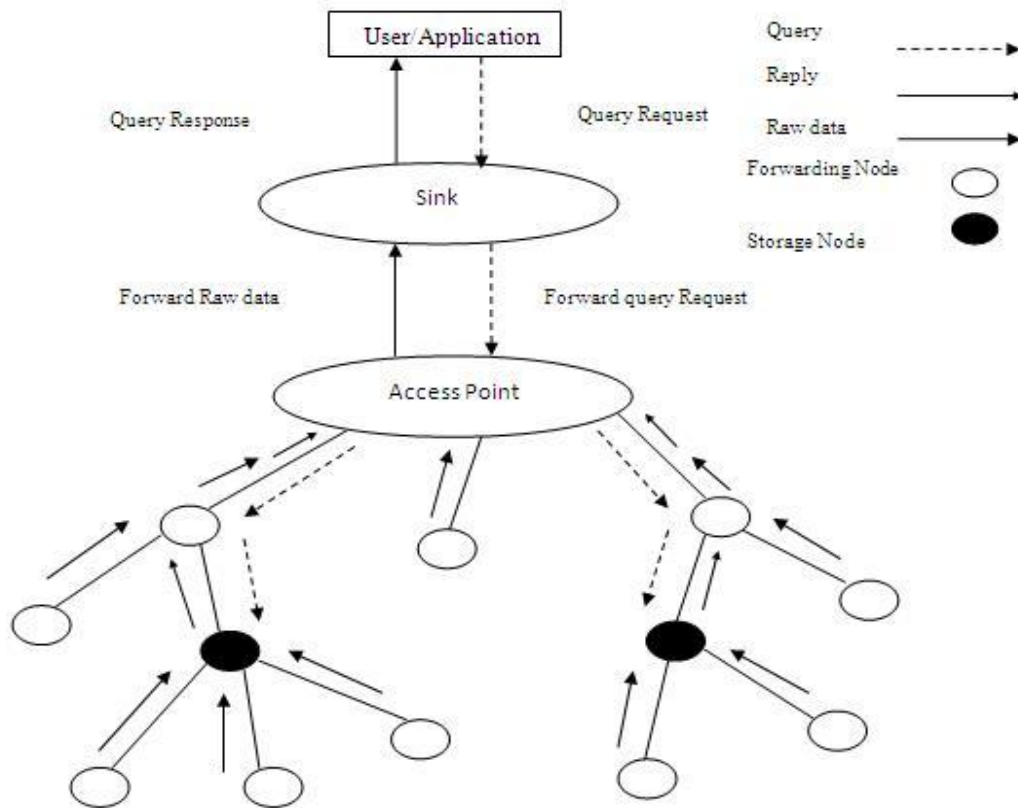

Figure 1: Data Access Model with Storage Nodes and Forwarding Nodes

A. **Access Point:** When the user fires the query on the sink, sink forward the query Request to the access point. Access point broadcast the query to sensor nodes. When the query arrived at storage nodes they forward the raw data back to the access point and then access point obtain the result and forward the data to the sink..
We first formally define two types of sensors (or nodes):

B. Storage nodes: These types of nodes have much larger storage capacity than normal sensor nodes. In the data access model as shown in fig.1, they store all the data received from other nodes or generated by themselves. Storage node does not send anything until queries arrive. According to the query specification, they receive the results needed from the raw data they are holding and then return the results back to the base station. The base station itself is considered as a storage node.

C. **Forwarding nodes:** These types of nodes are regular sensors and they always forward the data received from other nodes or generated by themselves along a path towards the sink. The outgoing data are kept intact and the forwarding operation continues until the data reach a nearest storage node. The raw data forwarding operation is independent of queries and there is no data   processing at forwarding nodes

IV.   Algorithm

The objective of this paper is the placement of storage sensor nodes on nodes in T such that total energy cost $\sum i \in Te(i)$ is minimized. In the case when there is limit on the number of storage sensor nodes i.e k ,to use ,that can be used to minimize the total energy cost, the problem is denoted with LIMITED.
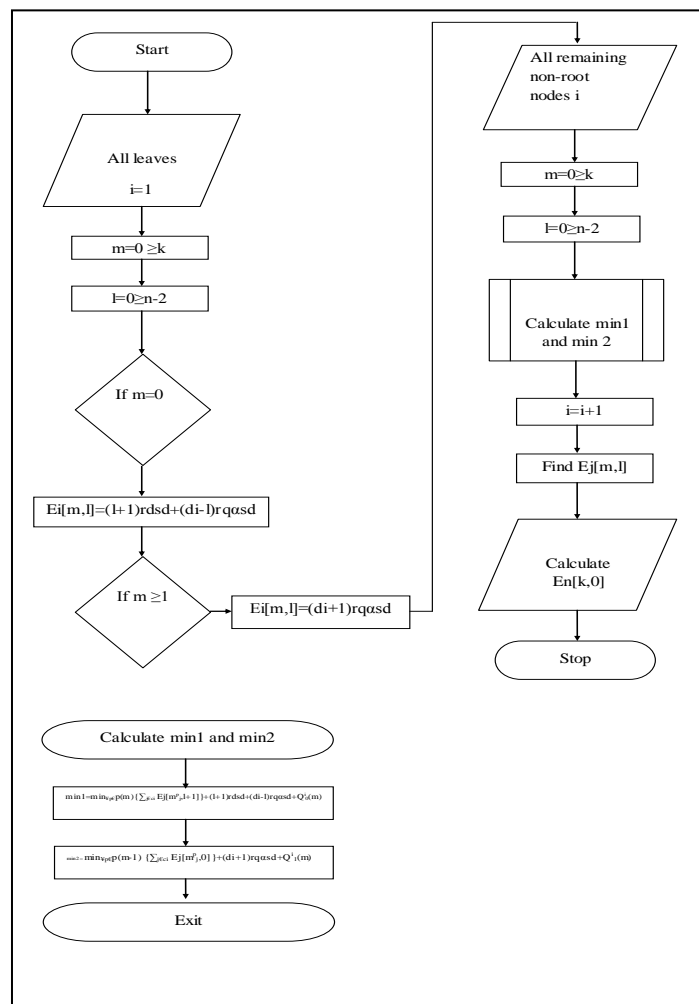

Figure 1: Algorithm flow chart for placing limited storage nodes

Here we consider the problem of storage node placement in sensor networks where the number of storage nodes is limited. Table 1 lists the notations, which we will be used in this algorithm.

k: total number of storage nodes (for LIMITED problem)
d_i : the depth of node i
c_i : the number of node i's children
r_d / s_d : rate of data generation / size of each data
r_q / sq : rate of user queries / size of each query message
n: total number of sensors
e(i) / E(i): energy cost of node i / energy cost of all the nodes in T_i
α: data reduction rate (query reply size / raw data size)
T_i : the subtree rooted at node i

Table 1: Notations

Let i be any node in the tree and Ti be the subtree rooted at i. We use |Ti | to denote the number of nodes in Ti . We define E(i) to be the cost of energy incurred at i per time unit, which includes, the raw data transfer cost from i to its parent if i is a forwarding node, the query diffusion cost. if i has storage nodes as its descendants, and the query reply cost if i is a storage sensor node or has a sensor storage descendant. To define E(i) mathematically we need to consider several possible cases.

Case I: i is a forwarding node and there are no storage nodes in Ti . All raw data generated by the nodes in Ti have to be forwarded to the parent of i and there is no query diffusion cost. So E(i) = |Ti |Rd Sd .

Case II: i is a storage node and there are no other storage nodes in Ti . The latest readings of all raw data generated by the nodes in Ti are processed at node i and the reduced reply size will be β|Ti |sd . Node i sends the reply to its parent when queries arrive. So E(i) = Rq β|Ti |Sd .

Case III. i is a storage node and there is at least one other storage node in Ti . In addition to the cost for query reply as defined in Case II, i also incurs a cost for query diffusion that is implemented by broadcasting to its children. So E(i) = Rq β|Ti |Sd + Bi Rq Sq .

Case IV. i is a forwarding node and there is at least one
Storage node in Ti . This is the case where all three types of cost (for query diffusion, raw data transfer, and query reply) are present. Among the |Ti |− 1 descendants of i, let d1 be the number of forwarding descendants without any storage nodes on their paths to i and d2 be the number of storage descendant's or forwarding descendants with at least one storage node on their paths to i . Obviously, d1 + d2 = |Ti |− 1. So
E(i) = (d1 + 1)Rd Sd + Bi Rq Sq + Rq βd2 Sd

The fig 1. Illustrates the flow chart for algorithm limited storage placement in sensor network. In this algorithm first we maintain a two dimensional (k+ 1) × (n − 1) table, Ei [m, l], at each node i, where $0 \leq m \leq k$ and $0 \leq l \leq n − 2$. In the algorithm, first we computes the Ei tables for all leaves i, then we compute the Ei tables for the remaining non-root nodes i, and last we compute the entry En [k, 0] for the root n. After all tables are constructed, the minimum energy cost of the communication tree with up to k number of storage nodes can be found in the entry En [k, 0]. Note that instead of constructing a table for the storage root n, we compute only the needed entry for n.

V.   Conclusion and Future Work

This paper considers the storage node placement problem in a sensor network. This paper introduces limited number of storage nodes in sensor network release the cost of sending all the raw data to a central place. In this paper, we examine how to place limited number of storage nodes to save energy for data collection and data query. This new model is much more simplified and implementable. We have tested it on different data sets available on internet using network simulator software. Our future work includes placement of unlimited number of storage nodes in sensor network to optimize query reply in a sensor network and to solve the storage node placement problem in terms of other performance metrics.

Acknowledgement

References

[1]. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, pages 56–67, New York, NY, USA, 2000. ACM Press

[2]. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw., 11(1):2−16, 2003.

[3]. S. Madden, M. J. Franklin, J. M. Hellerstein,and W.Hong TAG: a tiny aggregation service for ad-hoc sensor networks.SIGOPS Opererating System Review, 36(SI):131146,2002

[4]. P. Desnoyers, D. Ganesan, H. Li, M. Li, and P. Shenoy. PRESTO: A predictive storage architecture for Sensor networks. In Tenth Workshop on Hot Topics in Operating System, 2005

[5]. J. Newsome and D. Song. GEM: graph embedding for routing and data-centric storage in sensor networks without geographic information. In Proceedings of the 1st international conference on Embedded networked sensor systems, pages 76–88, New York, NY, USA, 2003. ACM Press

[6]. S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT, a geographic hash table. Mobile Networks and Applications, 8(4):427–442, 2003.

[7]. S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. SIGCOMMComputer Communication Review, 33(1):137–142, 2003

[8]. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan.Energy-efficient Communication Protocols for Wireless Microsensor Networks. In International Conference on SystemSciences, Maui, HI, January 2000.

[9]. P. Gupta and P. R. Kumar. The capacity of wireless networksIEEE Transactions on Information Theory, IT- 46(2):388–404March 2000.

[10]. D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks?SIGCOMM Comput. Commun. Rev., 33(1):143− 148, 2003.

[11]. E. J. Duarte-Melo and M. Liu. Data-gathering wireless Sensor networks: organization and capacity . Computer Networks (COMNET), 43(4):519–537, Nov. 2003.