Grenze International Journal of Engineering and Technology, Special issue

Role of Sensor Network to Save Energy for Storage Nodes

Dr.J Preetha[1] and Dr S. Deepajothi[2]
[1]Associate Professor, Department of Computer Science and Engineering, Muthayammal Engineering College, Rasipuram – 637 408, Tamilnadu, India
E-mail- psgpreetha@gmail.com
[2]Associate Professor, Department of Computer Science and Engineering, Gurunanak Institute of Technology, Ibrahimpatnam,R.R District– 501 506, India
E-mail: deepajothi.a@gmail.com

***Abstract*— In sensor networks the data storage and query reply are very important for archiving future information retrieval. Storage nodes are used to reduce heavy load and data collected while transmitting the data to the central place for archiving. The Storage nodes have much larger capacity than regular sensors .They do not send out anything until queries arrive. In query reply the user needs data, submits queries to the sink (base station), the sink diffuses queries to the storage nodes, based on the query description the storage nodes reply back to the sink. The improper placement of storage nodes maximizing energy cost associated with raw data transfers, query diffusion and query replies. The objective is to minimize the total energy cost for gathering data to the storage nodes with the proper placement of storage nodes and provide user-to-sink security.**

***Index Terms*— Sensor networks, storage nodes ,raw data transmission, query reply ,security.**

I. INTRODUCTION

A sensornet is a distributed sensing network comprised of a large number of small devices, each with some computational,storage and communication capability. Storage nodes are special sensors with much larger permanent storage (e.g., flash memory) and more battery power. Many senso network applications, e.g.,monitoring learning behavior of the children, environment sensing, etc., generate large amount of data continuously over a long period of time. Often, the large volume of data have to be stored somewhere for future retrieval and data analysis. One of the biggest challenges in these applications is how to store and search the collected data.Three main problems arise when data are stored in sensors. First, a sensor is equipped with only limited memory or storage space, which prohibits the storage of a large amount of data accumulated for months or years. Second, since sensors are battery operated, the stored data will be lost after the sensors are depleted of network and each sensor has to respond to query by transmitting data to power. Third, searching the data of interest in a widely scattered network field is a hard problem. Placing storage nodes is related to the sensor network applications. The query is the most important application for sensor networks since, in essence, sensor networks are about providing information of the environment to the end users. A user query may take various forms, e.g., "how many nodes detect vehicle traversing events?" and "what is the average temperature of the sensing field?" In this scenario, each sensor, in

Grenze ID: 01.GIJET.5.1.24

addition to sensing the nearby environment, is also involved in routing data for two network services:the raw data transmission to storage nodes and the transmission for query diffusion and query reply. Transmitting all the data to the sink or storing them on each sensor node locally based on two extrems. On one hand, data solely stored in the sink is beneficial to the query reply incurring no transmission cost, but the data accumulation to the sink is very costly. On the other hand, storing data locally incurs zero cost for data accumulation, whereas the query cost for becomes large because a query has to be diffused to the whole and each sensor has to respond to the query by transmitting data to the sink. The storage nodes serve as a buffer between the the sink and the sensor nodes. If the storage nodes are not placed properly in sensor network, it can be very complex with respect to various optimization metrics, such as bandwidth minimization on each link and lifetime of the network.

II. Related Work

In the literature, several schemes have introduced an intermediate tier between the sink and sensors. LEACH [11], a clustering based protocol that utilizes randomized rotation of local cluster base stations (cluster heads) to evenly distribute the energy load among the sensors in the network. LEACH, that minimizes global energy usage by distributing the load to all the nodes at different points in time and it also enhance lifetime of the system, but does not address the storage problem. In DCS [2], [10], [11],relevant data are stored by name at nodes within the sensornet, all data with the same general name (e.g, elephant sightings) will be stored at the same sensornet node. Queries for data with a particular name can then be sent directly to the node storing those named data without the flooding required in some data centric proposal. GHT system for DCS on sensornets GHT hashes keys into geographic coordinates ,and stored a key value pair at the sensor node geographically nearest the hash of its key.The two operation GHT support for Put (k,v) and Get(k).To avoid hashing key outside the sensornet,GHT requires approximate knowledge of boundaries. GEM [6],an infrastructure for node-to-node routing and data centric storage and information processing in sensor networks. In GEM, they constructed a labelled graph that can be embedded in the original network topology in an efficient and distributed fashion. In general, the DCS alleviated because the storage nodes collect the data generated within their proximities and avoid the heavy load on each link incurred by the long-distance data communication. In general the data centric storage schemes assume some understanding about data and extra cost is needed to forward data to the corresponding keeper nodes.

To facilitate data query, Ganesan [3] propose a multiresolution data storage system, where data are stored in a degrading lossy model, i.e., fresh data are stored completely while long-term data are stored lossily. The capacity of wireless networks analysed in [4],[8]. They scaled space and suppose that n nodes are located in a region of 1m2. Each node can transmit at w bits per second over a common wireless channel.They considered twtypesof networks,Arbitrary networks,where the node location,destination of sources and traffic demands are all arbitrary and Random networks,where the nodes and their destinations are randomly chosen. MULEs[9] pick up data from the sensors when in close range, buffer itand drop off the data to wireless access points. The disadvantage of this approach is, it increased latency because s nsors have to wait for a MULE to approach before the transfer can occur. PRESTO [7],a model driven predictive data management architecture for hierarchical sensornetworks.A proxy tier is introduced between sensor nodes and used terminals and proxy nodes can cache previous query responses. Proxy nodes in PRESTO have no resource constraints in terms of power,computation, storage, and communication. Capsule[5],an energy-optimized log-structured object storage system for flash memories that enables sensor applications to exploit storage resources in amultitude of ways.Capsule employs a hardwareabstraction layer that hides the vagaries of flash memories for the application and supports energy-optimized implementations of commonly used storage objects such as streams, files, arrays, queues and lists. storage nodes [1], are not placed properly in sensor network, it can be very complex with respect to various optimization metrics, suchas bandwidth minimization on each link and lifetime of the network.

III. Data Access Model

A sensor network is given with one special sensor identified as the sink (or base station)and many normal sensors,each of which generates (or collects) data from its environment. Users 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 is to let the sink have all the data. Then, any query can be satisfied directly by the sink. This requires each sensor to send its readings back to the sink immediately every time it generates

new data.Generally, transferring all raw data could be very costly and is not always necessary. Sensors to hold their data and to be aware of the queries, then raw data can be processed to contain only the readings that users are interested in and the reduced-size reply, instead of the whole raw readings, can be transferred back to the sink. This scheme is illustrated in Fig,1 where the black nodes,called storage nodes, are allowed to hold data. The sink diffuses queries to the storage nodes by broadcasting to the sensor network and these storage sensors reply to the queries by sending the processed data back., this approach reduces the raw data transfer cost (as indicated by the thick arrows in the figures)because some raw data transmissions are replaced by query reply (as indicated by the thin arrows). But this incurs an extra query diffusion cost,because of the improper placement of storage nodes.

1) We first formally define two types of sensors (or nodes). Storage nodes: This type of nodes have much larger storage capacity than regular sensors. In the data access model, they store all the data received from other nodes or generated by themselves. They do not send out anything until queries arrive. Based on the query description,they obtain the results needed from the raw data they are holding and then return the results back to the sink. The sink itself is considered as a storage node.

2) Forwarding nodes: This type of nodes are regular sensors and they always forward the data received from other nodes or generated by themselves along a path toward the sink. The outgoing data are kept intact and the forwarding operation continues until the data reach a storage node.

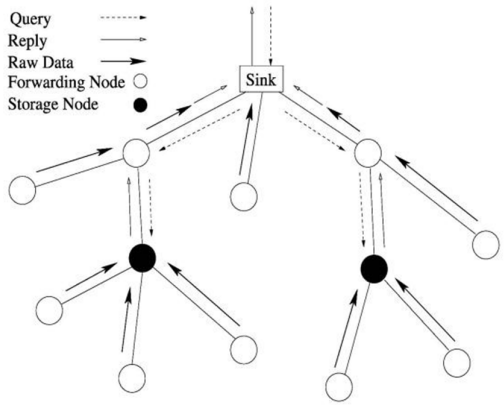

Fig 1.Data access model with storage nodes

Tree structure has been widely used in sensor networks.Although the communication tree may be broken due to link failure, this paper considers a common practice that only stable links are chosen to build the tree so that errors in transmission due to poor link quality can be reduced and the tree topology can be robust for a long time.For the storage node placement problem ,use the limited number of storage nodes in the fixed tree model.Within the fixed tree model the communication tree is formed.In the fixed tree model the storage nodes are placed after the formation of communication tree.

The security element involves mechanisms that avoid the compromise or damage of network resources, such as databases and e-mail messages. In the data access model,the user collects the raw data from the sink, between the user to sink not have the strengthen security.In this paper we use the Diffie-Hellman Key exchange method for security. Diffie– Hellman key exchange (D–H) is a specific method of exchanging keys. It is one of the earliest practical examples of Key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. The Diffie–Hellman exchange by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. In this paper, the goal is designing a data access model to minimize energy cost associated with raw data transfers, query diffusion and query replies and provide the security between the user to sink.

IV. LIMITED NUMBER OF STORAGE NODES

In this limited number of storage nodes within the fixed tree model,when there is limited number of storage nodes are placed,that can be used to minimize the energy cost.

A. Fixed Tree Model

In the fixed tree model, we first deploy regular sensors and construct a communication tree as usual. Based on the topology information, we select some of the regular sensors to be storage nodes. We can attach large flash memory to these selected sensors or replace them by more powerful storage nodes at the same locations. In the fixed tree model, the network builds a communication tree in which each node finds the shortest path to the sink by following the tree edges. Query messages are diffused from the sink to every storage node. For each storage node, it incurs an extra query diffusion cost along the path to its closest storage ancestorEach forwarding node sends its data to the first ancestor storage node on the path to the sink.

In LIMITED number of storage node placement is for the case of $\alpha r_q < r_d$. Since the number of storage nodes is limited, where to place them becomes a crucial problem. A bad placement strategy may hardly improve the performance.Basically, there is a trade-off between two trends. On the one hand, ifstorage nodes are close to the sink, i.e., at a high level in the tree structure, they can process more raw data, thus reduce the reply size from storage nodes to the sink. However, the sensor network spends much energy in transferring the raw data from low-level forwarding nodes to the storage nodes. On the other hand, if the storage nodes are far away from the sink, the raw data from their descendants can be processed earlier along the path toward the sink. However, storage nodes may cover only a few regular sensors, which leads to much raw data transferred to the sink without being processed.The algorithm 1 explains limited storage node placement.

TABLE I. SUMMARY OF NOTATIONS

r_d / s_d: rate of data generation / size of each data
r_q / s_q: rate of user queries / size of each query message
α: data reduction rate (query reply size / raw data size)
n: total number of sensors
k: total number of storage nodes (for LIMITED problem)
T_i: the subtree rooted at node i
d_i: the depth of node i
c_i: the number of node i's children
$e(i)$: energy cost of node i
$E(i)$: energy cost of all the nodes in T_i
λ / λ_s: density of regular sensors / storage nodes
e_{tr} / e_{re}: energy cost for transmitting / receiving a unit data

Algorithm 1. Place Limited Storage Nodes

1: for all leaves *i* do

2: for *m* = 0 to *k* do

3: for *l* = 0 to *n* - 2 do

4: if *m* = 0 then

5: $E_i[m,l] = (l+1)r_d\, s_d+(d_i - l)r_q \alpha s_d$

6: if $m \geq 1$ then $E_i[m,l]= (d_i +1)r_q \alpha s_d$

7: for all remaining nonroot nodes i, in Postorder,do

8: for $m = 0$ to k do

9: for $l = 0$ to n -2 do

10: min1 = min $\forall_p \in P(m) \{\sum_j \in Ci$

$E_j[m_j^{p,} l+1]\}+(l+1)r_d\, s_d+(d_i - l)$

$r_q \alpha s_d+Q_0^i(m)$

11: min2 = min$\forall_p \in P(m\text{-}1) \{\sum_j \in Ci$

$E_j[m_j^{p,} 0]\}+ (d_i +1)r_q \alpha s_d + Q_1^i(m)$

12: $E_i[m,l]$ = min{min1,min2}

13: E_n [k,0] = min $\forall_p \in P(k\text{-}1) \{\sum_j \in Cn$

$E[m_j^{p,} 0]\}+ r_q \alpha s_d + Q_1^i(m)$

14: return E_n [k,0]

Algorithm 1 maintains a two-dimensional (k+1) × (n-1) table, $E_i[m,l]$, at each node i, where 0 ≤m≤ k and 0 ≤ l ≤ n -2. Assume that a postorder traversal is done beforehand and that the depth of each node is computed beforehand. Both the postorder and the depths can be obtained in O(n) time. In the algorithm, lines 1-6 compute the E_i tables for all leaves i, lines 7-12 compute the E_i tables for the remaining nonroot nodes i, and lines 13 and 14 compute the entry E_n [k,0] for the root n. After all the tables are constructed, the minimum energy cost of the tree with up to k storage nodes can be found in the entry E_n [k,0]. Note that instead of constructing a table for the storage root n, we compute only the needed entry for n.

V. Conclusion

This paper considers the storage node placement problem in a sensor network. Introducing storage nodes into the sensor network alleviates the communication burden of sending all the raw data to a central place for data archiving and facilitates the data collection by transporting data from a limited number of storage nodes. In this paper, examine how to place storage nodes to save energy for data collection and data query and provide user to sink key exchange method for security.Using the limited number of storage nodes in the fixed tree model, place the storage nodes to minimize the energy cost.

References

[1] Bo Sheng, Member, IEEE, Qun Li, Member,IEEE, and Weizhen Mao,"Optimize Storage Placement in Sensor Networks", IEEE Transactions on Mobile Computing, Vol. 9, No. 10, October 2010

[2] C.T. Ee, S. Ratnasamy, and S. Shenker, "Practical Data-Centric Storage," Proc. Third USENIX Symp. Networked Systems Design and Implementation (NSDI '06), May 2006.

[3] D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R.Govindan, "Multiresolution Storage and Search in

Sensor Networks,"ACM Trans. Storage, vol. 1, no. 3, pp. 277-315, 2005.
[4] E.J. Duarte-Melo and M. Liu, "Data-Gathering Wireless Sensor Networks: Organization and Capacity," Computer Networks (COMNET), vol. 43, no. 4, pp. 519-537, Nov. 2003.
[5] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "CAPSULE:An Energy-Optimized Object Storage System for Memory-Constrained Sensor Devices," Proc. Fourth Int'l Conf. Embedded Networked Sensor Systems, 2006.
[6] J. Newsome and D. Song, "GEM: Graph Embedding for Routing and Data-Centric Storage in Sensor Networks without Geographic Information," Proc. First Int'l Conf. Embedded Networked Sensor Systems, pp. 76-88, 2003.
[7] M. Li, D. Ganesan, and P. Shenoy, "PRESTO: Feedback-Driven Data Management in Sensor Networks," Proc. Third USENIX Symp. Networked Systems Design and Implementation (NSDI '06), May 2006.
[8] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks,"IEEE Trans. Information Theory, vol. 46, no. 2, pp. 388-404, Mar.2000.
[9] R.C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," Proc. First IEEE Int'l Workshop Sensor Network Protocols and Applications (SPNA), May 2003.
[10] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin,and F. Yu, "Data-Centric Storage in Sensornets with GHT, a Geographic HashTable," Mobile Networks and Applications, vol. 8,no.4, pp. 427-442, 2003.
[11] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin,"Data-Centric Storage in Sensornets," SIGCOMM Computer Comm.Rev., vol. 33, no. 1, pp. 137-142, 2003.
[12] W.Heinzelman,A.Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," Proc. Int'l Conf. System Sciences, Jan. 2000