AN EFFICIENT DATA GATHERING IN WIRELESS SENSOR NETWORKS WITH MOBILE COLLECTORS

[1]K.Mathanagopal, [2]D.Venkadeshkumar
[1]PG Scholar, Department of ECE, Sri Guru Institute of Technology, Coimbatore
[2]Assistant Professor, Department of ECE, Sri Guru Institute of Technology, Coimbatore
[1]k.mathan005@gmail.com, [2]dvenkatdvk@gmail.com

Abstract: In this paper, we propose a new data-gathering mechanism for large-scale wireless sensor networks by introducing mobility into the network. Since the sensor nodes are equipped with small, often irreplaceable, batteries with limited power capacity, it is essential that the network be energy-efficient in order to maximize its lifetime. A mobile data collector, for convenience called an M-collector in this paper, could be a mobile robot or a vehicle equipped with a powerful transceiver and battery, working like a mobile base station and gathering data while moving through the field. An M-collector starts the data gathering tour periodically from the static data sink, polls each sensor while traversing its transmission range, then directly collects data from the sensor in single-hop communications, and finally transports the data to the static sink. For the applications with strict distance/ time constraints, we consider utilizing multiple M-collectors and propose a data-gathering algorithm where multiple M-collectors traverse through several shorter sub tours concurrently to satisfy the distance/time constraints. Our single hop mobile data gathering scheme can improve the scalability and balance the energy consumption among sensors. The proposed scheme is to increase the life time of the sensor network with the integrated gateway node. The IGN increases the life time of the network to integrate the multiple gateway nodes.

Keywords: Wireless Sensor Networks (WSNs), Data gathering, M-collector, Mobile data collector, Mobility, Clustering

1. INTRODUCTION

Wireless Sensor Networks have been noticed and researched in recent years. Wireless sensor networks (WSNs) have emerged as a new information gathering paradigm in a wide range of applications, such as medical treatment, outer-space exploration, battlefield surveillance, emergency response, etc. [2]. Sensor nodes are usually thrown into a large-scale sensing field without a preconfigured infrastructure. Before monitoring the environment, sensor nodes must be able to discover nearby nodes and organize themselves into a network. Most of the energy of a sensor is consumed on two major tasks: sensing the field and uploading data to the data sink. Energy consumption on sensing is relatively stable because it only depends on the sampling rate and does not depend on the network topology or the location of sensors. On the other hand, the data-gathering scheme is the most important factor that determines network lifetime. These networks are composed of hundreds or thousands of sensor nodes which have many different types of sensors [3]. Therefore, for a large-scale data centric sensor network, it is inefficient to use a single static data sink to gather data from all sensors. In some applications, sensors are deployed to monitor separate areas. In each area, sensors are densely deployed and connected, whereas sensors that belong to different areas may be disconnected. Unlike fully connected networks, some sensors cannot forward data to the data sink via wireless links. A mobile data collector is perfectly suitable for such applications. A mobile data collector serves as a mobile “data transporter” that moves through every community and links all separated sub networks together. The moving path of the mobile data collector acts as virtual links between separated sub networks. In this paper, we consider applications, where sensing data are generally collected at a low rate and is not so delay sensitive that it can be accumulated into fixed-length data packets and uploaded once in a while. To provide a scalable data-gathering scheme for large-scale static sensor networks, we utilize mobile data collectors to gather data from sensors. Specifically, a mobile data collector could be a mobile robot or a vehicle equipped with a powerful transceiver, battery, and large memory. The mobile data collector starts a tour from the data sink, traverses the network, collects sensing data from nearby nodes while moving, and then returns and uploads data to the data sink. Since the data collector is mobile, it can move close to sensor nodes, such that if the moving paths well planned, the network lifetime can be greatly prolonged. Here, network lifetime is defined as the duration from the time sensors start sending data to the data sink to the time when a certain percentage of sensors either run out of battery or cannot send data to the data sink due to the failure of relaying nodes. In the following, for convenience, we use M-collector to denote the mobile data collector. We have considered how to plan the data gathering tour of a single M-collector. However, for some large-scale applications, each data-gathering tour may take such a long time that a single M-collector may not be sufficient to visit the transmission ranges of all sensors before their buffers overflow. A possible solution to this problem is to allow some sensors to relay packets from other nodes to the mobile data collector. The length of each tour can be reduced. However, the drawback of using relay is that some relaying nodes may fail faster than others. To avoid unbalanced network lifetime, we will stay with the one-hop data-gathering scheme by utilizing multiple M-collectors. In Section II, we describe related works. In Section III, we describe our system model, and in Section IV, we present detailed operation of the proposed algorithm. Simulation results in Section V. Finally, we give concluding remark sin Section VI

2. RELATED WORK

In order to enhance the network lifetime by the period of a particular mission, many routing protocols have been devised. One of these is network clustering, in which network is partitioned into small clusters and each cluster is monitored and controlled by a node, called Cluster Head and also congestion avoidance can be made. In the sensor network, sensor node or M collector can communicate with the base station directly or through the cluster head, or through other relaying nodes. In a direct communication, each node communicates directly with the base station. When the sensor network is large, the energy for communicating with the base station is correspondingly large. Hence, some nodes far apart from the base station will quickly run out of energy [9]. The other scheme is the clustering; where the nodes are grouped into clusters and one node of the cluster send all gathered data from the nodes in its cluster to the base station. The LEACH (Low energy Adaptive Clustering Hierarchy) is a self-organizing and adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensor nodes[9], [10] LEACH includes a randomized rotation of the high energy cluster head position such that it rotates among the sensors. Efficient clustering algorithms for WSN have to satisfy several requirements, such as:

- Clusters should cover entire sensor field.
- Average cluster size should be as large as possible to maximize data aggregation efficiency.
- The clusters should be repeatedly reorganized to balance energy consumption among the nodes.
- Clustering overhead should be small.
- Clustering algorithm should be simple enough to be performed by low performance processor with small available memory space.

3. SYSTEM MODEL

Clustered structure of a network is very beneficial to Energy conservation as shown in . The benefit comes from the data aggregation of cluster heads. Aggregation efficiency increases as more data packets are aggregated. This benefit, however, is limited in multi-hop networks since cluster size is limited by the radio transmission range of the nodes. On the other hand, clustering overhead increases since clustering becomes more complex. The complexity comes mainly from the following two reasons: First, in multi-hop networks, it is difficult to re- cluster in a synchronized way as in single-hop networks. Second, when one cluster is reorganized, i.e. the role of a cluster head shift from one node to another, physical region the cluster head covers is also changed. This may necessitate reorganization of other clusters to satisfy the above requirements 1 and 2. These two requirements are for efficiency of the clusters. Requirement 3 says that the clustering overhead should be continually generated for fair energy consumption among the nodes. Moreover, if the nodes have mobility, clustering overheads will be far more increased. Thus, the benefit of clustering can be cancelled by the clustering overhead. In single-hop networks node mobility does not affect any network operation as long as the node does not move out of the transmission range of any other node. Among many of the previous researches, the network models for hierarchical protocols for WSNs are single-hop networks in [10-12] and those for flat routing protocols are multi-hop networks [13-14]. We consider for the following network and application model.

- A lot of sensor nodes are dispersed randomly on an interested region.
- Sink nodes are placed at some convenient places in or near the sensor field. Thus the sink nodes should have user interface or capabilities to communicate with remote users high powered radio or wired connection. The number of sink nodes is very small compared with the number of sensor nodes. Thus they can have special capability, battery with larger capacity or external power supply.
- The sensor nodes have limited processing and communication capabilities in order to satisfy the low cost condition. Thus very complex and/or energy consuming algorithm is difficult to be adopted.
- All the sensor nodes have the same constant transmission ranges.
- Users request data from the sensor network by disseminating query packets through the sink nodes. And, the data sensed from each node is gathered by sink nodes through cluster heads so that users can access it through the sink nodes.

3.1 The initial flooding process:

Routing information is flooded from the sink nodes. The procedure of each node to set the routing information for each sink node is similar to the distance vector algorithm. In the routing information packet, the number of hops to a specific sink node and the address of transmitting node are included. When a node receives routing information from a neighbor node, it increases the number of hops by
one and uses the number as its own number of hops to the sink node, and then, retransmit this information with its own address. When different number of hops is received from different neighbor nodes, the smallest number of them is used. If a node receives smaller number after ithas retransmitted routing information, the smaller number should be again retransmitted to correct the propagated errors. Through this procedure, each node can know its own number of hops to a specific sink and the address of the next hop node to the node.

3.2 Clustering:

The initial clustering occurs during the initial routing information distribution. In a routing information packet, energy state information of the transmitting node should be included. When a node has transmitted routing information, every neighbor of the node, except those who have previously transmitted the information, will retransmit the information. The node can gather information about the energy states of every neighbor node with the routing information packet, and compare them with its own energy state. When a node has found that it has the local maximum amount of energy, it becomes cluster head and broadcast a cluster head advertisement (CHAD) message to its neighbors. Before a node decides to be a cluster head, it has to wait for a sufficient time to gather the energy state information from all the neighbor nodes. The nodes that decided not to be a cluster head wait for a CHAD message from any other node. If a node waiting for a CHAD message cannot receive one for a pre-determined time period, it repeats the exchange procedure of energy state information with other non-affiliated nodes. This procedure is repeated until every node is affiliated with one cluster head. Any nonaffiliated node affiliates with the node whose CHAD massage it first receives.

4. PROPOSED METHODOLOGY

The proposed scheme is to increase the life time of the sensor network with the integrated gateway node. The IGN increases the life time of the network to integrate the multiple gateway nodes. Link setup method to avoid link failure.

4.1 IGCP — A INTEGRATED GATEWAY-NODE CONTROLPROTOCOL

Although the hierarchical structure is energy-efficient and great in data-aggregation as well as in-network processing is easy, the hierarchical-structure cannot be maintained for a long time and needs to be created again because of intense energy use in nodes like cluster head. The flat structure has easier multi-hop communication that the hierarchical-structure and allows even energy use of each node. This study suggests the IGN (Integrated Gateway Node) Algorithm to compensate the vulnerability of two structures. It is an Algorithm in which virtual gateway nodes consisting of several nodes like the cluster of hierarchical-structure routing protocol and allows the flat structure routing protocols between virtual nodes.

4.2 Gateway-Selection:

One cluster head has one gateway node to a sink node. A gateway node is selected by cluster head among the nodes which are one hop closer to the sink node right after the cluster head is elected. The gateway node may be or not be a cluster member of the cluster head which selects it as a gateway. A cluster head sends a gateway selection (GWS) message to a selected gateway node, and thus the selected node can know whose gateway it is. Query dissemination from a sink node and data gathering to a sink node is performed through cluster heads and gateway nodes.

4.3 Link setting:

Cluster head advertisement (CHAD) is largely classified into the busy status and static status. The busy status is a status in which IGN is operating as IGNH (Integrated Gateway Node Head) and the static status is a status in which IGN is operating as IGNM (Integrated Gateway Node Member) because there is no need of the IGNH role. The busy status occurs when different IGN's are created, adjacent IGN's are newly created, or the routing table of adjacent IGN's is changed. In the busy status, IGNM does not play the role of IGNM node but plays the IGN role only and always stays awaken. IGNH in the busy status has many roles thus it does not play any role for data communication but only works for Control Signal Communication. The static status occurs when IGNH does not have any control communication signals between IGNH's during the communication period of IGN. The communication period consists of Frame and Sync. Frame is the time for virtual nodes to communicate and Sync is the time for schedule re-configuration and error process. Sync is a lot shorter than Frame. IGNH does not have many roles in the static status and IGNH works as IGNM node. Once the initialization level is over and virtual nodes are created, it becomes the busy status of Gradient Level. The operations that require IGNM transport between virtual nodes or IGNH information take place. In the first busy status, it does not move on to the static status until a routing table is created.

4.4 Flooding level

Above all, once cluster formation has been completed, the Flooding process is not executed. Based on the level acquired through flooding, nodes piggyback their own levels in the data transmitted to update them. This level updating process is seen in Fig. In this figure, (a) indicates the levels of child nodes if the root (or parent) node equals level 1. This figure shows available connections between individual nodes and root or child nodes. However, in the event that the node with level 3 malfunctions, of child nodes detecting that, the node whose upstream connection has relied only on a connection with the faulty node finds out a node with the highest layer level, i.e., with the lowest level value among its adjacent nodes. This node, then, sets its own level to the result of adding 1 to the existing level. These processes are happen for a level update. The processes (a), (b), and (c) represent level dependent data transmission from usual low-layer sensor nodes to a high layer gateway node

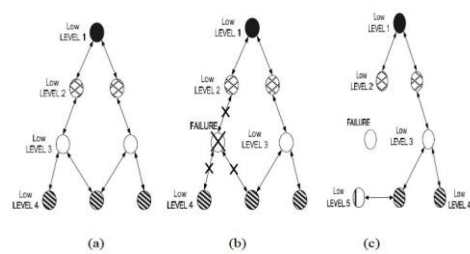

Figure 1: The level-updating procedure

5. SIMULATION AND RESULTS

The simulation parameters values used in our work are given below. This simulation results mainly consider throughput analysis and packet drop rate.

5.1 Throughput Analysis:

Throughput analysis for proposed and existing system is shown below in the fig. Throughput means The amount of data transferred from one place to another or processed in a specified amount of time.

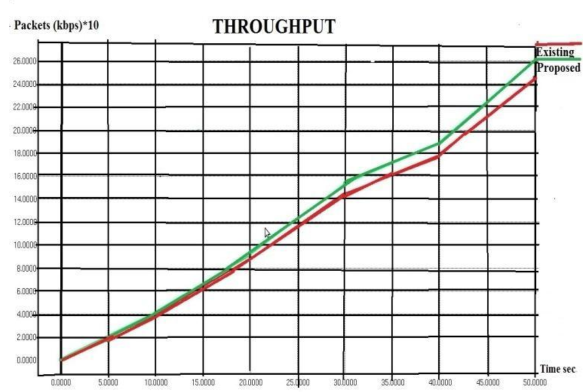

Figure 2: Throughput Analysis

In communication networks, such as Ethernet or packet radio, throughput or network throughput is the rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. The system throughput or aggregate throughput is the sum of the data rates that are delivered to all terminals in a network. Throughput is essentially synonymous to digital bandwidth consumption; it can be analyzed mathematically by means of queuing theory, where the load in packets per time unit is denoted arrival rate λ, and the throughput in packets per time unit is denoted departure rate μ Data transfer rates for disk drives and networks are measured in terms of throughput. People are often concerned about measuring the maximum data throughput in bits per second of a communications link or network access. A typical method of performing a measurement is to transfer a 'large' file from one system to another system and measure the time required to complete the transfer or copy of the file. The throughput is then calculated by dividing the file size by the time to get the throughput in megabits, kilobits, or bits per second. Throughput analysis performance is given for existing and proposed system. The system performance is given for both proposed and existing. In my project clearly understand that the throughput analysis of the network is more in proposed comparing with existing method.

5.2 Packet Drop Rate:

Packet drop rate is compared for both existing and proposed system and the graph is given below.

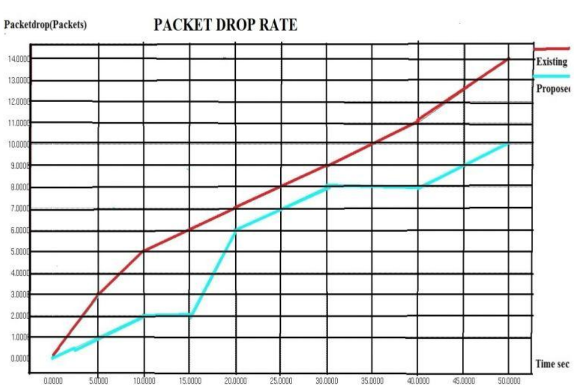

Figure 3: Packet Drop Rate

5.3 PACKET DROP RATE = NO OF PACKETS SENT – NO OF PACKETS RECEIVED.

Packet drop rate is mainly used for analyzing the data loss in the networks. Packet loss is the biggest enemy of getting good bandwidth. When a packet drops, the receiver must tell the sender to re-send it, which adds even more congestion, and in addition, transmission control protocol or internet protocol is designed to slow down in response to packet losses. The assumption in the protocol is that the packet loss is because the network is loaded; therefore dynamic adjustments take place to reduce the rate at which the packets are sent. Before eliminating packet loss as a problem when diagnosing a slow link, you really need to run a spray test to an echo port, which outputs UDP packets at the rate you expect the link to work, and check how many are dropped on the floor. Some implementations of ping have flags that you can use to adjust the size and rate of pings, which achieve the same effect. Tools such as these are normally only provided on Unix based systems, although there are windows utilities you can find of variable quality that try to do the same thing. Finally the packet drop rate is reduced in the proposed system while comparing to existing systems.

6. CONCLUSION

In this paper, we proposed a mobile data-gathering scheme for large-scale sensor networks. We introduced a mobile data collector, called an M-collector, which works like a mobile base station in the network. An M-collector starts the data gathering tour periodically from the static data sink, traverses the entire sensor network, polls sensors and gathers the data from sensors one by one, and finally returns and uploads data to the data sink. Our mobile data-gathering scheme improves the scalability and solves intrinsic problems of large-scale homogeneous networks. By introducing the M-collector, data gathering becomes more flexible and adaptable to the unexpected changes of the network topology In addition, data gathering by M collectors is perfectly suitable for applications, where sensors are only partially connected. For some applications in large scale networks with strict distance/time constraints for each data-gathering tour, we introduced multiple M-collectors by letting each of them move through a shorter sub tour than the entire tour. Proposed system is to increase the life time of the sensor network with the integrated gateway node. The IGN increases the life time of the network to integrate the multiple gateway nodes Link setup method is used to avoid the link failure.

REFERENCES

[1] Ming Ma, Yuanyuan Yang Felloe Ieee And Miaozhao "Tour Planning For Mobile Data-gathering Mechanisms In Wireless Sensor Networks "IEEE Transactions On Vehicular Technology, Vol. 62, No. 4, May 2013

[2] S. Tang, J. Yuan, X. Li, Y. Liu, G. Chen, M. Gu, J. Zhao, and G. Dai, "DAWN: Energy efficient data aggregation in WSN with mobile sinks," in Proc. IWQoS, Jun. 2010, pp. 1–9.

[3] F. Akyidiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless sensor network: A survey," Computer Networks, Vol. 38, No. 4, pp. 393–422, 2002.

[4] C. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," In Proceedings of the 1st international Conference on Embedded Networked Sensor Systems, Los Angeles, SenSys'03. ACM Press, New York, pp. 266–279, 05–07 November, 2003.

[5] S. Churchill, "Wireless bridge monitoring," MIT Technol. Rev., 2007. [Online]. Available: http://www.dailywireless.org/2007/08/13/ wirelessbridge- monitoring/

[6] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in Proc. ASPLOS, 2002, pp. 96–107.

[7] C. Ma and Y. Yang, "A battery-aware scheme for routing in wireless ad hoc networks," IEEE Trans. Veh. Technol., vol. 60, no. 8, pp. 3919– 3932, Oct. 2011.

[8] T. Small and Z. Haas, "The shared wireless infestation model—A new ad hoc networking paradigm (or wherethere is a whale, there is a way)," in Proc. ACM MobiHoc, 2003, pp. 233–244.

[9] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage and retrieval with an underwater sensor network," in Proc. ACM SenSys, 2005, pp. 154–165.

[10] A. Chakrabarty, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of a sensor network," in Proc. 2nd Int. Workshop IPSN, Apr. 2003, pp. 129–145.

[11] Farzad Tashtarian1, Mohsen Tolou Honary2], A. T. Haghighat 3, Jail Chitizadeh4,"A New Energy-Efficient Level-based Clustering Algorithm for Wireless Sensor Networks" published on IEEE Transaction 2007.

[12] Z. Khalid, G. Ahmed, N. M. Khan, and P. Vigneras, "A real-time energy-aware routing strategy for wireless sensor networks," Asia-Pacific Conference on Communications, Bangkok, Thailand, pp. 381–384, 2007.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Protocol for Wireless Microsensor Networks", in Proc. 33rd Annu. Hawaii Int. Conf. System Sciences, Hawaii, 2000.

[14] S. Lindsey, and C. S. Raghavendra, "PEGASIS: Power Efficient Gathering in Sensor Information Systems", in Proc. Aerospace Conf. Los Angeles, 2002.

[15] S. Lindsey, C. Raghavendra, and K. M. Siva lingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics", IEEE Trans. Parallel and Distributed Systems,vol. 13, no. 9, pp. 924-935, Sep. 2002.