Data Aggregation Tree Construction in Wireless Sensor Network

Zhiyi Lv
5080309948 F0803034
zhiyilv@gmail.com

***Abstract*—One important characteristic of wireless sensor network is it has severe energy constraints. To this end, many challenges arise such as capacity, lifetime, delay, robust,etc. Considering data of sensors is commonly correlate with its neighbor, data aggregation is one possible way to make difference in wireless sensor networks. In this paper, we will provide an algorithm of constructing tree and applying aggregation, and then evaluate its performance by total delay, and compare it with other algorithms. At last, we will briefly show what is interesting for further study.**

I. Introduction

Wireless sensor networks (WSNs) is constructed by a group of sensors linked by wireless medium to perform distributed sensing tasks. They can be used in many different areas, such as military, health, meteorology,etc. With rapid development in processor, memory, and radio technology, WSNs of small, inexpensive nodes become possible. There is no doubt that wireless sensor networks will play important role in collecting and processing information in diverse environments.

Although nodes ability increases(e.g. sensing, computation, and wireless communication), Sensor networks still facing some constrains. The most important one is limit energy. Due to this problem, it is impossible for all nodes transmitting their data to sink directly. People use multi-hop transmission to solve this problem. In a network with one stationary sink, nodes data is transmitted to sink by different nodes relay. The shortcoming is that burden of nodes near the sink will becomes numerous. That becomes the bottle neck of the capacity of WSN. In addition, it increases information delay and also reduce accuracy.

Data aggregation can be used in this condition. Because data generated by the sensors is often redundant and highly correlated, it is possible to find methods for combining data into high quality information. Data aggregation is defined as the process of aggregating the data from multiple sensors to e-liminate redundant transmission and provide fused information to the sink, and fusion usually happens at intermediate nodes and transmission of the aggregated data to the sink [1]. By data aggregation, the number of message packets transmitted in a network can greatly reduced, while information is little loss. Thus the amount of energy consumed can be reduced, and it extends lifetime of the network. Furthermore, since data aggregation makes data flows into sink in a more efficient way, data latency can be reduced, in other words, capacity of the network is increased.

With high energy efficiency and low information loss rate in consideration, It is critical to develop a wellperformance data aggregation algorithm. We will propose an algorithm by describing network architecture, the data aggregation mechanism and the underlying routing protocol step by step. And we will analysis the influence of the algorithm comparing with others.

II. Related works

There has been considerable research in the recent past on data aggregation in WSNs. Several significant algorithms has been put forward.

Data transmission algorithm is discussed under flat network model in [2,3]. These algorithms can be classified as push diffusion scheme. Sensor protocol for information via negotiation(SPIN) is presented in [2] and Intanagonwiwat et al. [3] developed directed diffusion which is an energy efficient data aggregation protocol. Their works focus on the flat networks in which each node plays the same role. An obvious shortcoming is that nodes near sink suffer excessive burden, and fast depletion of their battery will results in network breaking down. Thus heterogeneity is introduced to improve energy efficiency.

There are 4 major algorithms in data aggregation under hierarchical network model. Cluster based protocol is developed in [4,5,6,7], while [8] discussed chain based data aggregation. Works in[9,10] is based on tree structure.

Heinzelman et al. [4] propose an energy conserving cluster formation protocol, which is Low Energy Adaptive Clustering Hierarchy(LEACH). The LEACH protocol is distributed and sensor nodes organize themselves into clusters for data fusion. Cluster head in each cluster transmits the fused data from several sensors in its cluster to the sink. It is shown that LEACH delivers 10 times more data than MTE(minimum transmission energy routing) for the same number of node deaths. The assumption of LEACH is however too strong and an improved version called LEACH-C is proposed in [5]. Cluster formation is performed in a centralized manner by the sink. It improves the performance of LEACH by 20 to 40 percent. Another protocol Hybrid Energy Efficient Distributed Clustering Approach(HEED) is proposed by Younis et al. [6]. It forms efficient clusters for maximizing network lifetime. Simulation results show that HEED improves the network lifetime over LEACH and it achieves a well-distributed set of cluster heads. In [7], an aggregation scheme clustered diffusion with dynamic data aggregation(CLUDDA) is proposed. It is a hybrid new approach which combines directed diffusion [3] with cluster theory. In this protocol, the aggregation points are dynamic, any cluster head knows the query definition can perform data aggregation.

The protocol power efficient data gathering protocol for sensor information systems(PEGASIS) proposed in [8] is based on chain structure. It is introduced for condition that cluster head is far away from sensors. Nodes form a chain by greedy algorithm, and fuse their neighbor data with its own and transmits the fused data to its other neighbor along the chain. Its more energy efficient than LEACH. In [8], another two advanced protocols :binary chain based scheme and three-level chain based scheme are proposed and it is shown that their performance even better than PEGASIS. However, their assumptions are quite strong as well as PEGASIS. In designing tree based aggregation protocol, Ding et al. [9] have proposed an energy aware distributed aggregation (EADAT) and Tan et al. [10] have proposed a power efficient data gathering and aggregation protocol (PEDAP).

Sensor nodes are organized into a tree by broadcast information when initializing the algorithm. Data aggregation is performed at intermediate nodes along the tree and a concise representation of the data is transmitted to the root node. Both EADAT and PEDAP focus on lifetime of networks, however, EADAT quantize lifetime by number of alive sensors at the end of simulation time while PEDAP define lifetime as time until the death of last node.

In this paper, we construct a tree structure by the algorithm of EADAT, while applying some difference with it, and then propose one kind of scheduling. At last, we evaluate this whole process by its delay performance.

III. Constructing aggregation tree

A. preliminary

In our network, hundreds or even thousands of sensors are deployed, which means the number of nodes is pretty large. The function of the sensor network is to monitor some events such as fire in forest or gas leaking in coal mine. Data sensed is delivered to sink node which is considered to have infinite power and calculation ability. In contrast, the sensor is equipped with limited power supply, battery in common.

Besides the power supply, sensor has another three partssensing circuitry, digital processing and radio transceiver. We consider radio transceiver as the dominant energy consumer and energy for sensing or digital processing is negligible.[] However, if a sensor is in sleep mode(radio off), power consumption is also very low and it is treated as the most power saving state.

As we mentioned before, since adjacent sensors are monitoring the same environment, their data are highly correlated or should say similar. In some particular condition, such as monitoring the gas density in coal mine, only the highest value is needed. The redundant data should be pre-processed before they are transmitted. This is referred to as in-network processing.

This process often apply some aggregation function such as sum/avg/mean/max, etc. Different raw data are aggregated and delivered to the nodes nearer to sink and to be aggregated again, until they reach the sink. In a tree structure, the aggregation happens at each non-leaf node and process data from the subtree rooted at itself. This is called data-centric routing.

Here add another assumption which is reasonable and common used, that is all sensor nodes have the same transmission range and they work in a common channel.

B. heuristic algorithm

We illustrate the process of forming a tree structure step by step:

1) The process is initiated by sink node. It broadcasts a control packet. The packet has standard format which includes: ID, parent node, power, status, hop-Count to sink. Since sink is the root of the tree and it has infinite power, the packet is:(ID_s, -, ∞, status$_s$, 0);
2) A sensor v who got the packet will set its timer to a fixed time T_v^0. If the channel is idle, timer will begin to count down.

   T_v^0 indicates waiting time of v to broadcast its own message. It has relationship with v's residual power. The more power remains, the shorter time to wait. An reasonable assignment is $\frac{1}{power_v}$
3) When timer counts to zero, a new control packet which contains message of v will broadcast, while the hop-Count to sink will plus 1.
4) If a new packet is received by v during T_v^0, it indicates that another node, say u, has more residual power or is nearer to sink compare with v. Therefore v treats u as its parent node. If a third packet sent by t is received by v, it means t also can be v's parent node just less satisfied than u. Node v will record t in its list. It has a fixed buffer for recording.

   After receiving the second(or even more) control packet, v will back to step 2, reset T_v^0 and start counting down again.
5) Each of nodes will broadcast once. Since the link is symmetric, when v broadcasts its own control packet, its parent node u will receive it, then u mark itself as a non-leaf node. Otherwise, node will mark itself as leaf node.
6) The process will finish when all the nodes have broadcasted its own control packet. The tree is formed by each node recording its parent node.

There are some other problems should be considered.

1) When some changes occur to tree structure, for example, one node is moved, the original tree is not suit any longer. In this case, sink should reconstruct aggregation tree periodically. Since the construction process will cause power overhead, it shouldn't be done much frequently.
2) When a node has very low power remains, it is not certain when the node will stop work. If a node crashes

without any alarm, it will cause many problems such as loss of packets and wasted of energy. So when a node's residual power is less than a threshold P_{th}, it broadcasts a special control packet, say alarm packet, for a certain time to alarm its child nodes.

When a node v receives the alarm packet from its parent node, it starts to find a new parent node. It checks its parent list to find the second satisfied one, say t if exits. An exchanging message will send between v and t. If t is satisfied, the network is ok; or v will check the other one on the list.

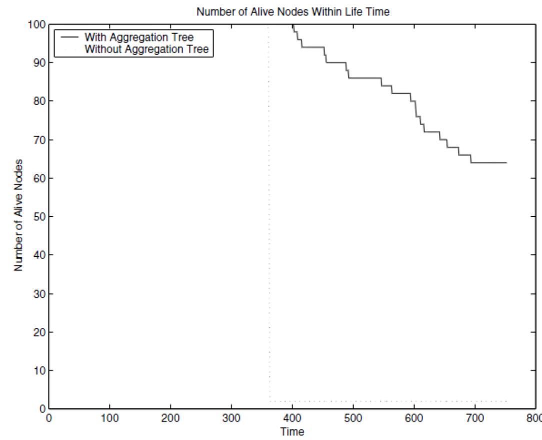
If v cannot find another parent node, v turns into danger state. It broadcasts its own message to find a parent node which is not recorded in its list. And if such node still not exists, it means that, v is isolated, it will not be included in the tree even if sink reconstruct the tree. v will send alarm packets to its child nodes.

3) Consider different nodes that share a same parent, for example node u, v share the same parent t. Since u and v are both in the interference area of t, and each of the nodes has the same transmission range, i.e. interference range, it is impossible for them to transmit data at the same time. If we use the similar method as construction of the tree, delay performance will be bad and with low energy efficiency. Therefore, a schedule should be made to minimize the total delay. The schedule will be discussed in section IV and its delay performance will be discussed in section V.

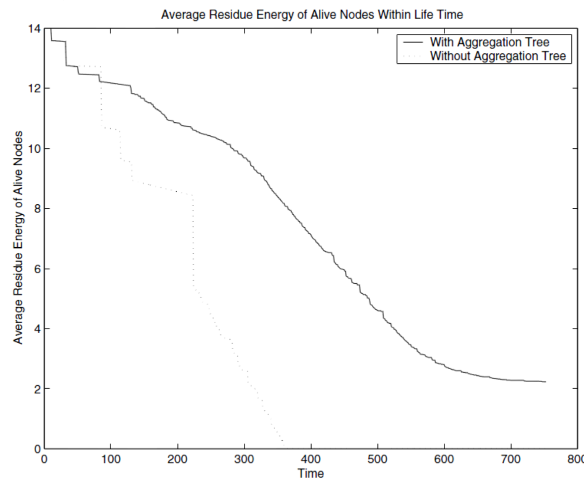
The heuristic method of constructing an aggregation tree has several nice features. Firstly, energy is concerned, nodes with more power will have more chance to be non-leaf node to play aggregation. Secondly, when the networks runs a long time, some nodes become battery low, it still guarantees high stability and reliability. In addition, the energy aware method can also be applied easily when data aggregating. Each node sets its timer according to is residual energy. As long as no interuption and when timer becomes zero, it transmits data to its parent node for aggregation and further transmission.

IV. PERFORMANCE

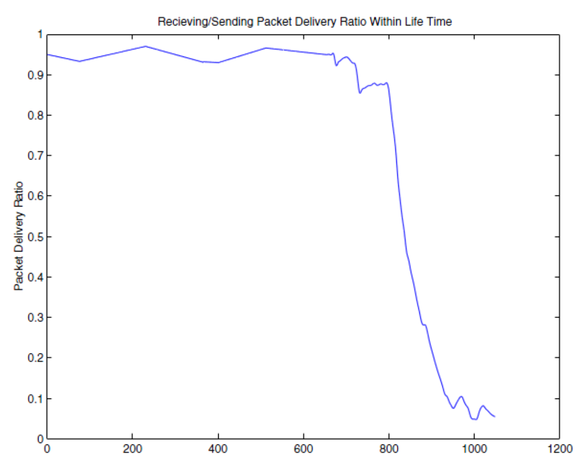
Since the procedure above is similar as the bibliography[9], our work is just adding some features to make the structure more robust, so the simulation in the paper is suitable here. Due to some personal reasons, I didn't finish the simulation completely. Here just quote the simulation results from [9]. The only difference is there is a little more overhead in the construction.



In this figure, we can see that, as time passes a certain point, the network suddenly break when without aggregation tree. When aggregation is introduced, although some nodes are died, the network still has its function.



This figure show that, after aggregation applied, as time passing by, residual energy of alive nodes decrease slower than that without aggregation. It is inuitively reasonable because less power is needed when transmitting the same amount of message.



The ratio is around 1 steadily when network is alive. This result shows that the protocol works stable and reliable.

V. Future work

A. Total delay

When we judging or designing the transmission scheme, we want to minimize to total transmission time, and that is goal of aggregation.

Let t_m denote the time when message m arrives at the sink, the problem can be denoted as: $minimize \sum t_m$. If weight of messages is concerned, the problem can be modified to $minimize \sum w_m t_m$, where $w_m$is weight of message m.

Obviously, $t_m$contains two parts, one is transmitting time, it equals to the hopcount to sink if we assume that it needs one time slot to transmit a packet from one node to another, denoted as h_m; another part is message's waiting time, i.e. delay in the network, denoted as D_m. Since if the aggregation tree is set up, $\sum h_m$is a constant, the problem can be interpreted as: $minimize \sum D_m$

Here should mentioned that, the total delay which is the accumulated delay of all the messages is in concern, while delay of single or part of messages is not meaningful. An interesting example is shown in table I in [11] that the optimal schedule is totally different when consider delay in different ways.

B. Optimal scheduling

In bibliography[11], lower bound of total delay is given, and it is also proved that the bound is tight. However, the optimal scheduling is not given in the paper, authors just use an example of binomial tree to prove that the bound is reachable. One of the future work is how to apply the method to general tree structure, or to analysis whether it is able to generalize the method.

C. Simulation

In this paper, we just quoted the simulation results from existing work. The most obstcle when we simulate is how to generate the event. What is its feature. There are several choices of distribution of event happening as well as nodes. Another future work is to compare the reliability of distributions in our scenario, building the test bed and evaluate the algorithm by ourselves.

VI. Conclusion

From the above, we presents an efficent and robust way to construct an aggregation tree. The structure can be well maintained after a long time and can be reconstructed periordically. Not only the construct procedure is proposed, the maintaining method is also presented to keep the structure available. The simulation shows that the aggregation algorithm make the network performs better.

Acknowledgment

The phase III of the project is mainly based on bibliography [9] and [11]. We evaluate the algorithm by quoting the simulation from exsiting work. And the scheduling part is not expanded in detail, instead it is just listed in the futrue work. Since the bibliography [11] doesn't presents a general method of scheduling, it is worth to pay more effort to finding it out.

And finally, best wishes and may you have a good summer vacation!

References

[1] Ramesh Rajagopalan and Pramod K. Varshney, *Data aggregation techniques in sensor networks: A survey*, IEEE Communications Surveys Tutorials, 2006.

[2] J. Kulik, W.R. Heinzelman and H. Balakrishnan, *Negotiation-based protocols for disseminating information in wireless sensor networks*, Wireless Networks, March 2002.

[3] C. Intanagonwiwat, R. Govindan and D. Estrin, *Directed Diffusion: A Scalable and robust communication paradigm for sensor networks*, MobiCOM, August 2000.

[4] W. R. Heinzelman, *Application-specific protocol architectures for wireless networks*, PhD Thesis, Massachusetts Institute of Technology, June 2000.

[5] W .R. Heinzelman, A.P . Chandrakasan, and H. Balakrishnan, *An application-specific protocol architecture for wireless microsensor networks*, IEEE Trans Wireless Communications, October 2002

[6] O. Younis and S. Fahmy, HEED: a hybrid, energyefficient, *distributed clustering approach for ad hoc sensor networks*, IEEE Transactions on Mobile Computing, Dec 2004

[7] S. Chatterjea and P.Havinga, *A Dynamic data aggregation scheme for wireless sensor networks*, Proc. Program for Research on Integrated Systems and Circuits, Nov. 2003.

[8] S. Lindsey, C. Raghavendra, and K.M. Sivalingam, *Data gathering algorithms in sensor networks using energy metrics*, IEEE Trans. Parallel and Distributed Systems,September 2002,

[9] M. Ding, X. Cheng and G. Xue, *Aggregation tree construction in sensor networks*, IEEE Vehicular Technology Conference, October 2003

[10] H. O. Tan and I. Korpeoglu, *Power efficient data gathering and aggregation in wireless sensor networks*, SIGMOD Record, December

[11] Changhee Joo, Jin-Ghoo Choi, and Ness B. Shroff, *Delay Performance of Scheduling with Data Aggregation in Wireless Sensor Networks*, IEEE INFOCOM 2010