

A Distributive Approach for Data Collection and Consignment Balancing In Wireless Sensor Networks via Mu-Mimo

¹ P.Fayeeza Begum & ² K. Ramesh Rao

¹M.Tech Dept of CSE, ALITS College, Affiliated to JNTUA, AP, India .

²Associate Professor, Dept of CSE, ALITS College, Affiliated to JNTUA, AP, India

Abstract— A three-layer framework is proposed for mobile data collection in wireless sensor networks, which includes the sensor layer, cluster head layer, and mobile collector (called Sensor) layer. The framework employs distributed load balanced clustering and dual data uploading, which is referred to as LBC-DDU. The objective is to achieve good scalability, long network lifetime and low data collection latency. At the sensor layer, a distributed load balanced clustering (LBC) algorithm is proposed for sensors to self-organize themselves into clusters. In contrast to existing clustering methods, our scheme generates multiple cluster heads in each cluster to balance the work load and facilitate dual data uploading. At the cluster head layer, the inter-cluster transmission range is carefully chosen to guarantee the connectivity among the clusters. Multiple cluster heads within a cluster cooperate with each other to perform energy-saving inter-cluster communications.

Through inter-cluster transmissions, cluster head information is forwarded to Sensor for its moving trajectory planning. At the mobile collector layer, Sensor is equipped with two antennas, which enables two cluster heads to simultaneously upload data to Sensor in each time by utilizing multi-user multiple-input and multiple-output (MU-MIMO) technique.

Key Words: Load balanced clustering algorithm; Data Collection; Mobility; Dual antenna.

I. INTRODUCTION

Proliferation of the implementation for low-cost, low power, multifunctional sensors has made wireless sensor networks, a major data collection model for extracting local measures of benefit. sensors are generally densely deployed and randomly scattered over a sensing field and left unattended after being deployed, which makes it difficult to recharge or replace their batteries. After sensors form in to autonomous organizations, those

sensors near the data sink typically deplete their batteries much faster than others due to more relaying traffic. When sensors around the data sink exhaust their energy, network connectivity and coverage may not be guaranteed. as sensing data in some applications are time-sensitive, data collection may be required to be performed within a specified time frame. Therefore, an efficient, large-scale data collection scheme should aim at good scalability, long network lifetime and low data latency. Several approaches have been proposed for efficient data collection in the literature. The first category is the enhanced relay routing [4], [7], [8] in which data are relayed among sensors. Besides relaying, some other factors, such as load balance, schedule pattern and data redundancy, are also considered. The second category organizes sensors into clusters and allows cluster heads to take the responsibility for forwarding data to the data sink. Clustering is particularly useful for applications with scalability requirement and is very effective in local data aggregation since it can reduce the collisions and balance load among sensors.

The third category is to make use of mobile collectors to take the burden of data routing from sensors. minimizing energy consumption on the forwarding path does not necessarily prolong 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 more 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. Based on these observations, in this paper, we propose a three-layer mobile data collection framework, named Load Balanced Clustering and Dual Data Uploading (LBC-DDU). A distributed algorithm to organize sensors into clusters, where each cluster has multiple cluster heads. Algorithm



balances the load of intra-cluster aggregation and enables dual data uploading between multiple cluster heads and the mobile collector. Second, multiple cluster heads within a cluster can collaborate with each other to perform energy efficient inter-cluster transmissions. A mobile collector with two antennas (called SenCar in this paper) to allow concurrent uploading from two cluster heads by using MU-MIMO communication. The SenCar collects data from the cluster heads by visiting each cluster. It chooses the stop locations inside each cluster and determines the sequence to visit them, such that data collection can be done in minimum time.

2. SENCAR LAYER

The sensor layer is the bottom and basic layer. For generality, we do not make any assumptions on sensor distribution or node capability, such as location-awareness. Each sensor is assumed to be able to communicate only with its neighbours, i.e., the nodes within its transmission range. During initialization, sensors are self-organized into clusters. Each sensor decides to be either a cluster head or a cluster member in a distributed manner. Sensors with higher residual energy would become cluster heads and each cluster has at most M cluster heads, where M is a system parameter. For convenience, the multiple cluster heads within a cluster are called a cluster head group (CHG), with each cluster head being the peer of others. The algorithm constructs clusters such that each sensor in a cluster is one hop away from at least one cluster head. The FT designed to static WSNs, and there is no predefined topology to transfer the data from the sensor nodes to sink. Here, all the sensor nodes directly communicate with the sink or simply forwards the data packets to the one-hop neighbour nodes and finally reach to the sink. The existing methods have limitation such as delay, node failure, data redundancy and large amount of energy utilization, since it is using flooding, gossiping, direct communication, etc., to communicate between the nodes. It is the main drawback of this topology and not recommended to mobile WSNs. Upon the arrival of SenCar, each CHG uploads buffered data via MU-MIMO communications and synchronizes its local clocks with the global clock on SenCar via acknowledgement messages. Finally, periodical re-clustering is performed to rotate cluster heads among sensors with higher residual energy to avoid draining energy from cluster heads. Fig 2 describes the SenCar layer with LBC-DDU.

The cluster head layer consists of all the cluster heads. As a fore mentioned, inter-cluster forwarding is only used to send the CHG information of each cluster to SenCar,

which contains an identification list of multiple cluster heads in a CHG. Such information must be sent before SenCar departs for its data collection tour. Upon receiving this information, SenCar utilizes it to determine where to stop within each cluster to collect data from its CHG. To collect data as fast as possible, SenCar should stop at positions inside a cluster that can achieve maximum capacity[7][8]. In theory, since SenCar is mobile, it has the freedom to choose any preferred position.

However, this is infeasible in practice, because it is very hard to estimate channel conditions for all possible positions. Thus, we only consider a finite set of locations. To mitigate the impact from dynamic channel conditions, SenCar measures channel state information before each data collection tour to select candidate locations for data collection. We call these possible locations SenCar can stop to perform synchronized data collections polling points. In fact, SenCar does not have to visit all the polling points. Instead, it calculates some polling points which are accessible and we call them selected polling points. In addition, we need to determine the sequence for SenCar to visit these selected polling points such that data collection latency is minimized. Since SenCar has pre-knowledge about the locations of polling points, it can find a good trajectory by seeking the shortest route that visits each selected polling point exactly once and then returns to the data sink.

2.1 MU-MIMO

MU-MIMO can greatly speed up data collection time and reduce the overall latency. Another application scenario emerges in disaster rescue. For example, to combat forest fire, sensor nodes are usually deployed densely to monitor the situation. These applications usually involve hundreds of readings in a short period (a large amount of data) and are risky for human being to manually collect sensed data. A mobile collector equipped with multiple antennas overcomes these difficulties by reducing data collection latency and reaching hazard regions not accessible by human being[1][2][10]. Although employing mobility may elongate the moving time, data collection time would become dominant or at least comparable to moving time for many high-rate or densely deployed sensing applications. In addition, using the mobile data collector can successfully obtain data even from disconnected regions and guarantee that all of the generated data are collected.

3. LOAD BALANCED CLUSTERING

The essential operation of clustering is the selection of cluster heads. To prolong network lifetime, we naturally expect the selected cluster heads are the ones with higher

residual energy. Hence, we use the percentage of residual energy of each sensor as the initial clustering priority. Assume that a set of sensors, denoted by $S = \{s_1; s_2; \dots; s_n\}$, are homogeneous and each of them independently makes the decision on its status based on local information. After running the LBC algorithm, each cluster will have at most $M (>1)$ cluster heads, which means that the size of CHG of each cluster is no more than M . Each sensor is covered by at least one cluster head inside a cluster. The LBC algorithm is comprised of four phases: (1) Initialization; (2) Status claim; (3) Cluster forming and (4) Cluster head synchronization.

3.1 Initialization phase

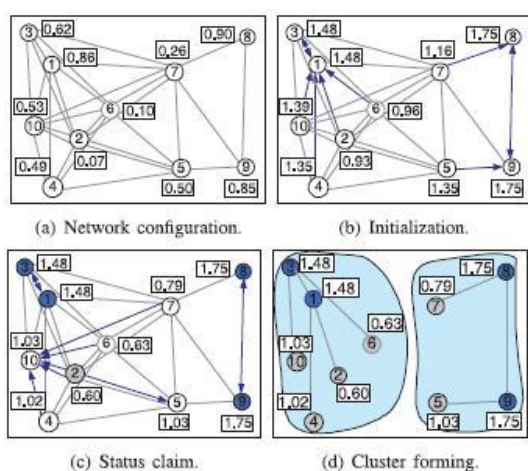


Fig 1 LBC Algorithm $M=2$

It implies that once s_i successfully claims 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. s_i sets its priority by summing 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. Fig. 1b depicts the initialization phase of the example, where M is set to 2, which means that each sensor would pick one neighbor with the highest initial priority as its candidate peer.

3.2 Status claim

In the second phase, each sensor determines its status by iteratively updating its local information, refraining from promptly claiming to be a cluster head. 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, t_h and t_m ($t_h > t_m$), which

enable a sensor to declare itself to be a cluster head or member, respectively, before reaching its maximum number of iterations. During the iterations, in some cases, if the priority of a sensor is greater than t_h or less than t_m compared with its neighbours, it can immediately decide its final status and quit from the iteration. We call this process self-driven status transition. Also, s_i will announce In the initialization phase, each sensor acquaints itself with all the neighbours in its proximity. If a sensor is an isolated node (i.e., no neighbour exists), it claims itself to be a cluster head and the cluster only contains itself. Otherwise, a sensor, say, s_i , first sets its status as “tentative” and its initial priority by the percentage of residual energy. Neighbours 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 .

its current candidate peers to be cluster heads by broadcasting a packet including an ID list, which is referred to as the peer-driven status transition.

3.3 Cluster forming

The third phase is cluster forming that decides which cluster head a sensor should be associated with. The criteria can be described as follows: for a sensor with tentative status or being a cluster member, it would randomly affiliate itself with a cluster head among its candidate peers for load balance purpose. In the rare case that there is no cluster head among the candidate peers of a sensor with tentative status, the sensor would claim itself and its current candidate peers as the cluster heads. Cluster members that receive this message switch to the initialization phase to perform a new round of clustering.

3.4 Synchronization among cluster heads

To perform data collection by TDMA techniques, intra cluster time synchronization among established cluster heads should be considered. The fourth phase is to synchronize local clocks among cluster heads in a CHG by beacon messages. First, each cluster head will send out a beacon message with its initial priority and local clock information to other nodes in the CHG. Then it examines the received beacon Messages to see if the priority of a beacon message is higher. If yes, it adjusts its local clock according to the timestamp of the beacon message. In our framework, such synchronization among cluster heads is only performed while SenCar is collecting data. Because data collection is not very frequent in most mobile data



gathering applications, message overhead is certainly manageable within a cluster[2][14].

4 Cluster Head Layer-Connectivity Among CHGs

Multiple cluster heads in a CHG coordinate among cluster members and collaborate to communicate with other CHGs. Hence, the inter-cluster communication in LBC-DDUs essentially the communication among CHGs. By employing the mobile collector, cluster heads in a CHG need not to forward data packets from other clusters. Instead, the inter-cluster transmissions are only used to forward the information of each CHG to SenCar.

The inter-cluster organization is determined by the relationship between the inter-cluster transmission range R_t and the sensor transmission range R_s . Clearly, R_t is much larger than R_s . It implies that in a traditional single-head cluster, each cluster head must greatly enhance its output power to reach other cluster heads.

However, in LBC-DDU the multiple cluster heads of a CHG can mitigate this rigid demand since they can cooperate for inter-cluster transmission and relax the requirement on the individual output power.

Cluster heads in a CHG as multiple antennas both in the transmitting and receiving sides such that an equivalent MIMO system can be constructed[6][7][8]. The self-driven cluster head in a CHG can either coordinate the local information sharing at the transmitting side or act as the destination for the cooperative reception at the receiving side. Each collaborative cluster head as the transmitter encodes the transmission sequence according to a specified space-time block code (STBC) to achieve spatial diversity. Compared to the single-input single-output system, that a MIMO system with spatial diversity leads to higher reliability given the same power budget. An alternative view is that for the same receive sensitivity; MIMO systems require less transmission energy than SISO systems for the same transmission distance. Therefore, given two connected clusters, compared with the single-head structure, in which the inter-cluster transmission is equivalent to a SISO system, the multi-head structure in LBC-DDU can save energy for inter-cluster communication.

4.1 MU-MIMO Uploading

We jointly consider the selections of the schedule pattern and selected polling points for the corresponding scheduling pairs, aiming at achieving the maximum sum of MIMO uplink capacity in a cluster. We assume that SenCar utilizes the minimum mean square error receiver

with successive interference cancellation (MMSE-SIC) as the receiving structure for each MIMO data uploading. Based on this receiver, the capacity of a 2×2 MIMO uplink between a scheduling pair $\delta_a; \delta_b$ where h_a and h_b are two 2×1 channel vectors between cluster heads a and b and SenCar at \sim , respectively, P_t is the output power of a sensor for transmission range R_s , and N_0 is the variance of the back-ground Gaussian noise. The MMSE-SIC receiver first decodes the information from a , treating the signals of b as the interference. Then, it cancels the signal part of a from the received signals. The remaining signal part of b only has to contend with the background Gaussian noise. Once the selected polling points for each cluster are chosen, SenCar can finally determine its trajectory. The moving time on the trajectory can be reduced by a proper visiting sequence of selected polling points. Since SenCar departs from the data sink and also needs to return the collected data to it, the trajectory of SenCar is a route that visits each selected polling point once. This is the well-known travelling salesman problem (TSP). Since SenCar has the knowledge about the locations of polling points, it can utilize an approximate or heuristic algorithm for the TSP problem to find the shortest moving trajectory among selected polling points, e.g., the nearest neighbor algorithm[10][11]

5. CONCLUSION

In this paper, we have proposed the LBC-DDU framework for mobile data collection in a WSN. It consists of sensor layer, cluster head layer and SenCar layer. It employs distributed load balanced clustering for sensor self-organization, adopts collaborative inter-cluster communication for energy-efficient transmissions among CHGs, uses dual data uploading for fast data collection, and optimizes SenCar's mobility to fully enjoy the benefits of MU-MIMO. Our performance study demonstrates the effectiveness of the proposed framework. The results show that LBC-DDU can greatly reduce energy consumptions by alleviating routing burdens on nodes and balancing workload among cluster heads, which achieves 20 percent less data collection time compared to SISO mobile data gathering and over 60 percent energy saving on cluster heads. We have also justified the energy overhead and explored the results with different numbers of cluster heads in the framework. Finally, we would like to point out that there are some interesting problems that may be studied in our future work. The first problem is how to find polling points and compatible pairs for each cluster. A discretization scheme should be developed to partition the continuous space to locate the optimal polling point for each cluster. Then finding the compatible pairs becomes a matching problem to achieve optimal overall

spatial diversity. The second problem is how to schedule MIMO uploading from multiple clusters. An algorithm that adapts to the current MIMO-based transmission scheduling algorithms should be studied in future.

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] 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.
- [4] 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.
- [5] 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.
- [6] 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. 10, pp. 3004–3018, Oct. 2010.
- [7] 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.
- [8] 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.