CAC: Context Adaptive Clustering for Efficient Data Aggregation in Wireless Sensor Networks

Guang-yao Jin and Myong-Soon Park †

Dept. of Computer Science and Engineering, Korea University
Seoul 136-701, Korea
{king, myongsp}@ilab.korea.ac.kr

Abstract. Wireless sensor networks are characterized by the widely distributed sensor nodes which transmit sensed data to the base station cooperatively. However, due to the spatial correlation between sensor observations, it is not necessary for every node to transmit its data. There are already some papers on how to do clustering and data aggregation in-network, however, no one considers about the data distribution with respect to the environment. In this paper a context adaptive clustering mechanism is proposed, which tries to form clusters of sensors with similar output data within the bound of a given tolerance parameter. With similar data inside a cluster, it is possible for the cluster header to use a simple technique for data aggregation without introducing large errors, thus can reduce energy consumption and prolong the sensor lifetime. The algorithm proposed is very simple, transparent, localized and does not need any central authority to monitor or supervise it.

1 Introduction

In the case when a sensor network is sensing simple data, such as the temperature of a room exposed to sunlight, it can be assumed that there will be several regions where measured temperature is similar under a specific tolerance. As the sun moves from East to West, those areas are going to change slowly as well. The problem is that certain regions would be in the adjacent area of different clusters, thus those adjacent cluster headers would have to send some overlapped data to the base station for correct data aggregation. This will generate more network traffic and energy consumption.

For example, in Figure 1, there are two clusters and two different temperature regions. Each cluster header only needs to transmit one data to the base station after data aggregation. After a certain time period, the temperature distribution will change as shown in Figure 2. In this case, existing approaches of data aggregation (e.g., work out an average as proposed in [5]) would produce two representative temperatures per region (e.g., a1 and a2 in region a) in order to maintain the high data correlation in a cluster which has a localized property. Thus, each cluster header will transmit two data representatively and together transmit four. However, some sub-regions such as

† Corresponding Author: Myong-Soon Park; E-mail: myongsp@ilab.korea.ac.kr

(a2) and (b1) have the same localized property and produce identical representative data correspondingly. It is unnecessary to consume energy to send the same data item twice (e.g., a2 and b1).

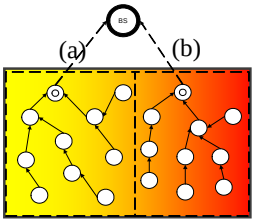

Fig. 1. Initial temperature Distribution.

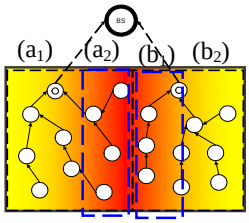

Fig. 2. The temperature distribution was changed.

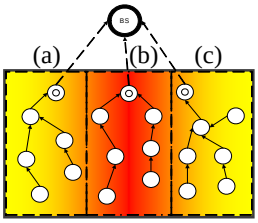

Fig. 3. After adaptive clustering was performed

Our approach is to use an adaptive re-clustering algorithm in order to faithfully represent the physical reality. As shown in Figure 3, since the temperature distribution (i.e., context) changed, the two regions (or clusters) adaptively change into three regions (clusters) without a centralized (global) component processing, and each region send a representative data separately, thus there are all together three data need to be send. In this way, our approach could produce correct representative data (better represents the physical reality), reduce energy consumption of sensor nodes and prolong sensor network life.

This paper is organized as follows. In section 2, the related work is presented. Section 3 proposes our algorithm, and Section 4 discusses the simulation results of our algorithm. Section 5 concludes the paper with future work.

2 Related Work

There have been some researches on clustering in wireless sensor networks as discussed in [1], [2], [3], [6], [7]. In [1] the cluster heads are identified once during network deployment by a central controller. Also in [2] the clusters are formed during the actual physical deployment of the sensor networks which have to be planned by the network designers in advance. However, our proposed algorithm does not need a central controller and clustering is performed dynamically through the sensor network lifetime. [3] and [4] require the priori information of location and the initial sensor energy. However, in our approach such information is not needed to form and update the clusters dynamically. The LEACH protocol [6] for clustering and cluster-head determination was proposed, which goal is to organize clusters based on the energy level of sensor nodes and to re-circulate the elected cluster-header inside a cluster in order to save battery power of the nodes. However, the algorithm this paper proposed is concerned about organizing clusters based on the data they sense, i.e. geographically partitioning physical space into clusters of correlated data, and thus making data aggregation more effectively and faithfully represent the physical reality. An improved version of LEACH, which is called LEACH-C [7], has a set-up phase for initial cluster-head computation by the base station. However, our approach does not

require such initial step, since it concerns with forming the clusters based on their data output and the re-clustering is done by the network itself in a certain region.

In summary, the main feature of our algorithm compared with existing clustering algorithms is a very effective technique in sensing which is localized and does not require computations by some higher (central) entities and re-clustering is performed dynamically to keep high data correlation.

3 Context Adaptive Clustering (CAC)

As discussed in Section 1, data aggregation can be performed efficiently and correctly when data from different sensors are highly correlated. But if the collected data change over time due to the change of actual physical environment, data correlation must be changed correspondingly. One of the CAC goals is to maintain the high data correlation within a cluster and therefore save more sensor energy.

3.1 Assumption

In this paper we assume that the sensed data is changing smoothly over a long time period, and the data has a regional property, i.e. in one specific area data is similar, so cluster can be formed and data can be aggregated by the cluster header.

3.2 The Proposed Algorithm

At initial deployment, how many geographic clusters a sensor network region can be partitioned is manually determined to form initial clusters. Adaptive re-clustering is performed locally by header nodes using the proposed approach recursively until stability is achieved. The stability is defined as a state that re-clustering is ended in the network and all clusters have correlated data, based on the threshold tolerance.

More formally, a set of the correlated nodes in a cluster, e.g. $\sigma(d,\delta)$, is defined to determine whether a node d_i belongs to the cluster or not. Its input parameters are the aggregated data value (d) and the tolerance parameter (δ). If a node d_i is in $\sigma(d,\delta)$ (i.e., $d_i \in \sigma(d,\delta), i = 1,2...n$, where the cluster size is n), node d_i belongs to the cluster. That is, a node belongs to the cluster if and only if its output data is equal to the aggregated value (d) or bounded around it given the tolerance parameter δ . When the time passes by, the data distribution changes smoothly. If there are p nodes whose data do not belong to $\sigma(d,\delta)$ (i.e., $d'_j \notin \sigma(d,\delta), j = 1,2...p$, where $p < n$), data of these nodes are separated into m ($1 \le m \le p$) different ranges. Then the cluster header needs to use $m+1$ data to represent the whole data correctly assuming a simple algorithm, such as computing an average, for data aggregation. $D = \{d_j \mid d_j \notin \sigma(d,\delta), j = 1,2...p\}$ is used to indicate the p nodes, and $D'_k, k = 1,2,...,m$

indicates the m different ranges, here $D = \sum_{1}^{m} D'_k$. The worst case takes place when $(m = p) \& (p = n-1)$. In this case there are no benefits from data aggregation. Given a threshold M, when m becomes bigger than M, the cluster header initially generates the new headers list $H = \{h_k \mid h_k \in D'_k, k = 1,2,..., m\}$ which contains new possible cluster header candidates and then the re-clustering will be performed.

Loop
Gathering data from d_1 to d_n ;
Compute D ;
Compute m and D'_k ;
	If $m \geq M$ then
	Generate H ;
	Broadcast re-clustering ($\sigma(d,\delta), H$);
	Else
	Do data aggregation

Fig. 4. Algorithm to decide when to start re-clustering

Read its data as d' ;
If $my\_id \in H$ then
// consider itself as header and broadcast
Broadcast re-clustering ($\sigma(d',\delta)$, Null);
	Else
	If $d' \in \sigma(d,\delta)$ then
	Join the new cluster;
	Forward re-clustering ($\sigma(d,\delta), H$);

Fig. 5. Algorithm to decide to join the cluster

Algorithm in Figure 4 is executed by the current cluster header to decide whether to initiate re-clustering or not. If re-clustering is needed, cluster header forms a list of nodes that fall out of its current range and broadcasts the list to the nearby nodes. Nearby nodes examine their current sensor output and decide whether they join one of the new possible clusters or stay in their current one as explained in Figure 5.

A node receiving the re-clustering command either considers itself as a cluster header and re-broadcasts the command, or joins a new cluster based on their current sensor output and forwards the re-clustering command to other nodes in its vicinity. If a node is neither a new cluster header nor interested in joining a new cluster, it would disregard the received command, which would stop and bound the algorithm to a certain geographic region.

4 Experimental Results

To validate the energy efficiency by reducing data items that have to be transmitted to satisfy the tolerance parameter, we have simulated both the LEACH and our proposed mechanism in NS2. In our experiments, we used a 100-node network where nodes were randomly distributed between (0, 0) and (100, 100). The radio model adopted in this experiment is based on [8]. The function used for data aggregation was computing an average of the data received from the nodes in a cluster, which was computed based on the nodes' location in the area and the current location of the data source using the Euclidian distance formula for the 2D-plane, considering the fact that the node which is the nearest to the data source would have a maximum output of 100

and the farthest one would have a minimum output of 0, and the outputs of other distributed sensor nodes are evenly spaced between 0 and 100 accordingly.

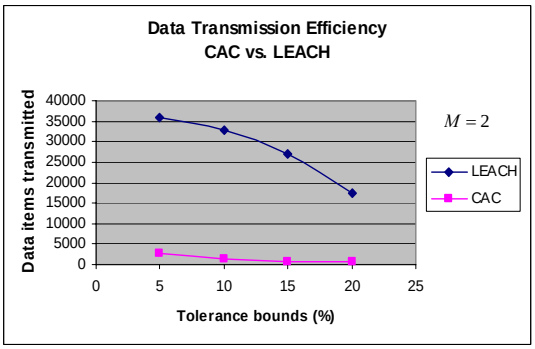

Fig. 6. Simulation results for 100 nodes

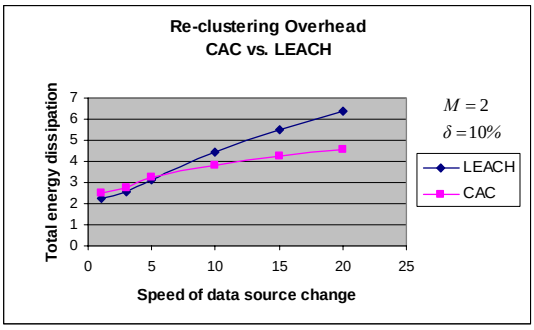

Fig. 7. Total energy dissipation of CAC vs. LEACH

For each time increment, data is being sent from all the clusters. A cluster header would send the aggregated data item and may still send a data item from the sensors that do not fit in the tolerance criterion, but however, in this case, re-clustering would be performed. Results of the simulation are given in Figure 6 for the case of 100 sensor nodes. Our approach in Figure 6 shows, in terms of how many data items are to be transmitted, a 31.1-fold improvement for 10% tolerance and a 29.78-fold improvement for 20% tolerance compared with LEACH. Thus, by having fewer transmissions, implicitly power consumption is reduced without affecting reliability or availability.

In case of the overhead imposed by CAC, according to our assumptions of slowly changing data, re-clustering will not be performed very frequently, so the overhead can be considered negligible. In case the data changes very rapidly, such an overhead for re-clustering would be significant, thus other solutions than ours would be more suitable. Results of such experiment are given in Figure 7.

For smooth changing data, CAC reduces the number of data items that have to be transmitted and thus enhance the network lifetime by requiring less data transmissions.

5 Conclusions and Future work

In this paper, an energy-efficient algorithm is proposed that can generate clusters in a sensor network for data aggregation and adapt the clusters by performing re-clustering depending on the data changes caused by the environment changes. Resulting clusters have similar data that can be easily aggregated by the cluster header without introducing large error in the aggregated data output using an efficient, computable and inexpensive algorithm, and thus power consumption is reduced. Further more, CAC is localized, that is, a re-clustering is performed regionally and independently from both other clusters in some other area and the central authority, such as base station.

We would like to focus our future work on how to decide the optimal M for CAC, since the parameter M directly affects the performance of CAC.

6 Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD) (KRF-2005-211-D00274).

References

1. Jason Tillet, Raghuveer Rao and Ferat Sahin, “Cluster-Head identification in ad hoc sensor networks using particle swarm optimization”. Proc. of the IEEE International. Conference on Personal Wireless Communication, 2002.2.
2. Wei-Peng Chen, Jennifer C. Hou and Lui Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks“. Proc. 11th IEEE Conf. on Network protocols, 2003.
3. Kostuv Dasgupta, Konstantinos Kalpakis and Parag Namjoshi, “An efficient clustering-based heuristic for data gathering and aggregation in sensor networks”. Proc. of the IEEE Wireless Communications and Networking Conference, March 16-20, 2003.
4. Konstantinos Kalpakis, Koustuv Dasgupta, and Parag Namjoshi, “Maximum lifetime data gathering and aggregation in wireless sensor networks". Proc. of the IEEE International Conference on Networking (ICN'02), Atlanta, Georgia, August 26-29, 2002. pp. 685-696.
5. J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate Aggregation Techniques for Sensor Databases”. Proc. of the 20th International Conference on Data Engineering, 2004
6. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication protocol for wireless microsensor networks”. Proc. of the 33rd Hawaii International Conference on System Sciences, 2000.
7. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”. IEEE Transactions on Wireless Communications, vol.1, no.4, October 2002.
8. Jain-Shing Liu; Lin, C.-H.P. “Power-Efficiency Clustering Method with Power-Limit Constraint for Sensor Networks”. Proc. of the 2003 IEEE International, 9-11 April 2003.