

# Exploiting Spatial Correlation Towards an Energy Efficient Clustered AGgregation Technique (CAG)

SunHee Yoon and Cyrus Shahabi

Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

{sunheeyoo, shahabi}@usc.edu

**Abstract**—In Wireless Sensor Networks (WSN), monitoring applications use in-network aggregation to minimize energy overhead by reducing the number of transmissions between the nodes. We note that nearby sensor nodes monitoring an environmental feature (e.g., temperature or brightness) typically register similar values. In this paper, we propose Clustered AGgregation (CAG), which is a mechanism that reduces the number of transmissions and provides approximate results to aggregate queries by utilizing the spatial correlation of sensor data. The result is guaranteed to be within a user-provided error-tolerance threshold. While a query is disseminated to the network, CAG forms clusters of nodes sensing similar values. Subsequently, only one value per cluster is transmitted up the aggregation tree. We use mathematical models and simulations with synthetic and empirical data to evaluate the efficiency-correctness tradeoff of CAG. Our simulation shows that with highly correlated sensor reading and 10% error threshold, CAG can save the communication overhead by as much as 70.9% over TAG while incurring a modest 1.7% error in result.

## I. INTRODUCTION

In WSN, in-network query processing is a common way to minimize communication by increasing path sharing as in Directed Diffusion [6], TinyDB [12], and Cougar [20]. TinyDB, the landmark in-network query processing system for WSN, has a fixed set of query operators supported by a query processor. Alternately, directed Diffusion allows users to define their own in-network aggregation operators. A tree-based routing is used in Tiny AGgregation (TAG) [12], while a data-centric routing is used in Directed Diffusion [6].

Structural [9] and habitat [13] monitoring, the most popular applications of WSN to date, can be efficiently implemented by using those in-network aggregation systems. They enable a user to issue a query to be flooded to the network to build data forwarding and aggregation plans. Such flooding-based systems can be made more energy efficient by exploiting the spatial correlation in sensor data.

Allowing for an approximate result, and not requiring an exact answer, enables designing energy-efficient mechanisms to compute in-network aggregates. Approximate results can be used in an interactive setting in which users may first ask for a rough picture of regional data before they decide to drill-down further [3]. In this scenario, not every sensed data is required

This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), IIS-0238560 (PECASE), IIS-0324955 (ITR) and IIS-0307908, and unrestricted cash gifts from Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

to compute the synopsis. Both energy efficiency and accuracy are important in time-critical monitoring. In many systems, however, higher accuracy comes at a higher energy cost.

Olston *et al.* designed an adaptive bounded-width filter which trades precision for communication overhead [16]. Jain *et al.* tried to minimize resource usage under precision requirement by designing a prediction system using Dual Kalman Filter (DKF) [7]. As such, sophisticated prediction schemes can be incorporated in WSN to prevent unnecessary data transmission.

Techniques such as LEACH [5], TEEN [14], APTEEN [15] use hierarchical clusters and routing to save energy. Patten *et al.* studied correlation between data spatial coherence and routing efficiency using lossless compression [17].

PREMON [4] and TiNA [18] are similar to CAG. PREMON forms clusters based on a prediction model while CAG forms clusters using real-time sensor values. TiNA exploits temporal correlation in sensor data while CAG takes advantage of spatial correlation to form clusters. Deshpande *et al.* proposed a data acquisitional method based on statistical model [2]. Unlike CAG, their study does not take into account packet losses in the network; neither do they use clusters.

CAG exploits *semantic broadcast* [19] in order to reduce the communication overhead by leveraging *spatial correlation*, the characteristic of the data distribution. CAG achieves efficient in-network storage and processing by allowing a unified mechanism between query routing (networking) and query processing (application). Instead of gathering and compressing all the data (lossless algorithm), CAG generates synopsis by filtering out insignificant elements in data streams (lossy algorithm) to minimize response time, storage, computation, and communication costs.

Although environmental attributes such as temperature, light, and sound could be correlated over large distances, there has been no in-network aggregation algorithm exploiting spatially correlated sensor data aiming at both efficiency and precision challenges. To the best of our knowledge, CAG is the first in-network aggregation algorithm exploiting spatial correlation, which trades a negligible quality of result (precision) for a significant energy saving. CAG achieves this by focusing on a few representative values rather than a large number of redundant data. With denser sensor deployment, there will be even higher data correlation, which increases CAG's efficiency and precision.

CAG is a lossy clustering algorithm because CAG uses

```

function Query.Received:
  if  $((CR - CR \times \tau) \leq v_{ij} < (CR + CR \times \tau))$ {
    clusterhead = FALSE;
    broadcast query  $Q$ ;
  } else{
    CR = MR;
    clusterhead = TRUE;
    broadcast query  $Q$ ;
  }

function Response.Received:
  enqueue response packet  $R$  to the buff;

function Epoch.Fired:
  if clusterhead
    broadcast aggregate(buff + MR);
  else if size(buff) > 0
    broadcast aggregate(buff);

```

TABLE I

PSEUDOCODE OF THE CLUSTERED AGGREGATION ALGORITHM

only the sensor values from the clusterheads to compute the aggregate. Values of new clusterheads differ from the values of parent clusterheads by at least the user-provided threshold. We systematically quantify the impact of spatial correlation in sensor data on in-network aggregation by using the CAG algorithm.

The rest of this paper is organized as follows. Section II describes and analyzes the CAG algorithm. Section III presents three different data sets, evaluation metrics, and simulation results. Finally, Section IV concludes the paper.

## II. CAG: A LOSSY CLUSTERING TECHNIQUE FOR AGGREGATION

In this Section, we describe the CAG algorithm and analyze its efficiency and accuracy.

### A. The CAG Algorithm

CAG branches out from TAG for further energy saving by using spatial correlation of data. TAG, a landmark system performing in-network aggregation, requires every node to participate in aggregation while CAG requires only representative values to participate in aggregation. The prevalence of spatial correlation in environmental phenomena makes it possible for CAG to ignore redundant data and quickly generate an overview of the data distribution.

The CAG algorithm operates in two phases: query and response. During the query phase, CAG forms clusters when TAG-like forwarding tree is built using a user-specified error threshold  $\tau$ . In the response phase, CAG transmits a *single* value per cluster. CAG is a lossy clustering method; only the clusterheads contribute to the aggregation.

A *user-provided error threshold*,  $\tau$ , is used while building clusters. Each node decides to join a cluster based on *Clusterhead sensor Reading (CR)* and *My local sensor Reading (MR)*; if  $MR < CR \pm CR \times \tau$ , then the sensor is included in the same cluster. That is why  $\tau$  is interchangeable with

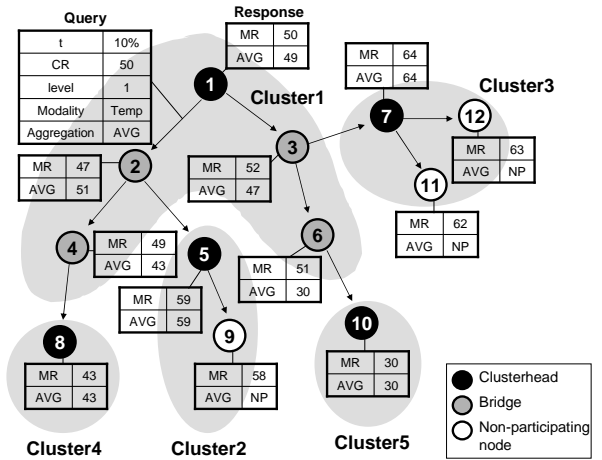


Fig. 1. An example execution of the Clustered AGgregation algorithm.

a *user-provided error-tolerance threshold*. Table I shows the pseudocode of the CAG algorithm.

For encoding a query, CAG augments the TAG syntax with a *threshold*  $\tau$ . The user initiates CAG by specifying a query  $UQ = \langle QueryID, O_i, \tau \rangle$  to be injected into the network with a threshold  $\tau$  for the monitoring attribute  $O_i$ . Subsequently, the base station broadcasts the query packet  $Q = \langle UQ, ParentID, MyID, level, CR \rangle$ , where *level* is the depth of the current node in the forwarding tree. Note that CR is included in the query to be compared with each MR when it is received by a node. Clusters are formed when the forwarding tree is built.

Once all the nodes receive the query packet, the response phase starts. At the end of each epoch, only clusterheads transmit packets with the following tuple:  $R = \langle ParentID, ChildrenID, MR, CR \rangle$ . Bridge nodes do not contribute their sensor readings to the aggregate, but they are required to *bridge* the segments of the forwarding tree.

Fig. 1 shows an example execution of the CAG algorithm. Tables in Fig. 1 describe main attributes embedded in a query packet (connected to the link) and a response packet (connected to the node). The sensor reading 50 of root node (node 1) automatically becomes the first CR. When a node (node 2) receives the query, it determines if its local value MR is within the calculated tolerable error range  $CR \pm CR \times \tau$  (between 45 and 55 in this example). If MR is within this range, this node is included in the same cluster with the clusterhead (cluster 1). Otherwise, this node becomes a new clusterhead by assigning MR to CR. In our example, sensor readings for nodes 2, 3, 4, and 6 are within the range, so they stay in the same cluster. Clusterhead selection policy applied in CAG is such that the first node becomes a clusterhead if its sensor reading is outside of the tolerable error range for CR. Where there is a high disparity among monitored sensor readings (node 5, 7, 8, 10), the new clusters are formed.

In response phase, only the clusterheads contribute their sensor readings to the aggregate. Node 2 in this example is a bridge node; it performs in-network aggregation (without

Variable	Description
$n_i$	node $n_i$ has location $x_i$ and $y_i$
$v(n_i)$	sensor reading at node $n_i$
$D_{ij}$	distance between nodes $n_i$ and $n_j$
$\tau$	user specified error threshold
$N$	total number of sensor nodes
$N_q$	E(Number of Query)
$N_c$	E(Number of Clusterheads)
$N_b$	E(Number of Bridges)
$N_p$	E(Number of participating nodes) $N_p = N_c + N_b$
$N_{bc}$	E(Number of transmissions) $N_{bc} = N_q + N_p$

TABLE II  
VARIABLE DEFINITION

including its local reading) with values received from nodes 4 and 5.

CAG forwarding tree is based on dynamic environmental phenomena, so the clusterhead may change for every query and response cycle. This prevents clusterheads from becoming energy-draining bottlenecks.

### B. Formalization and Analysis

In this Section, we try to formally quantify 1) the energy saving of CAG in terms of number of transmissions and 2) the accuracy of result in terms of  $\tau$  and distance. Table II defines the variables used in the following analysis.

To simplify the analysis, we assume that the nodes are placed in a two dimensional grid, and all nodes are within radio range  $R$  performing lossless communication. Also, we assume that no node participates in multiple clusters. We do not include bridge nodes to simplify our analysis. That is,  $N_p = N_c$  in Section II-B.1.

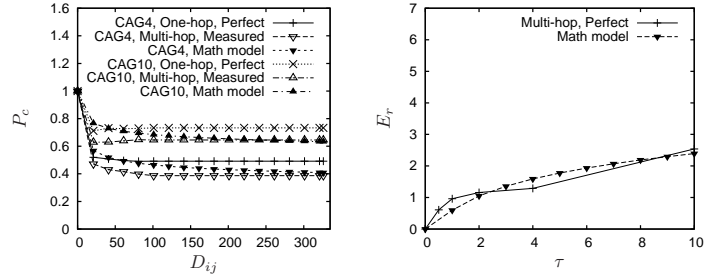
1) *Number of transmissions with CAG:* We first model the performance of CAG in terms of *number of transmissions* with uncorrelated i.e., *i.i.d.* random data normalized to  $[0, 1]$ . To calculate the number of transmissions, we need to compute the expected number of clusters because only clusterhead nodes transmit values. Let  $P_u$  be the probability that a node is in the same cluster with the root node in a scenario with uncorrelated data.  $P_u$  is given by:

$$P_u = P(|v(n_j) - v(n_i)| < \tau v(n_i) | v(n_i)) \\ = \begin{cases} 2\tau v(n_i) & \text{if } v(n_i) \leq \frac{1}{1+\tau}, \\ 1 - (1 - \tau)v(n_i) & \text{if } v(n_i) > \frac{1}{1+\tau}. \end{cases} \quad (1)$$

Because the root node is always in the root cluster, and all other nodes ( $N - 1$  nodes) are in the same cluster with a probability  $P_u$ , the expected size of the root cluster is as follows:

$$E(\text{size of root cluster}) = 1 + (N - 1)P_u \quad (2)$$

Hence, the expected number of clusters can be calculated from



(a) Simulation model (one-hop perfect reliability and multi-hop measured reliability) and Mathematical model of  $P_c$  described in Equation (4) ( $\tau = 4, 10\%$ ) (b) Simulation with data generated using the parameter 7H and Mathematical model of the relative error  $E_r$  described in Equation (10) ( $C_1 = C_2 = 1$ )

Fig. 2. Spatial correlation model of CAG: (a)  $P_c$  and (b)  $E_r$ . CAG with  $\tau = n\%$  is termed  $CAGn$ .

the Equations (1) and (2) as follows:

$$N_c = 1 + N - E(\text{size of root cluster}) \\ = 1 + N - (1 + (N - 1)P_u) \\ = N - (N - 1)P_u \quad (3)$$

$N_q = N$  for both TAG and CAG. In TAG, all the nodes respond to the query. Thus,  $N_{bc} = 2N$  for TAG. For CAG,  $N_{bc} = N_q + N_c = 2N - (N - 1)P_u$  with uncorrelated data.

Now we model the performance of CAG when the sensor data is spatially correlated based on each sensor's geographical location in a two dimensional space.

Spatially correlated sensor data is modeled and plotted in Fig. 2(a). Only the results with  $\tau = 4\%$  and  $10\%$  are presented due to space constraint. To mathematically model the function of CAG, we computed Equation (1) using the CAG algorithm under 1) single-hop, perfect reliability (same assumptions as in the second paragraph of Section II-B) and 2) multi-hop, measured reliability.

Although we assume single-hop lossless topologies in this analysis, the prediction based on our mathematical model matches results from simulations with realistic node density, packet losses, multiple hops, and data correlation (Fig. 2(a)).

Let  $P_c$  be the probability that any two nodes are in the same cluster in a scenario with correlated data.  $P_c$  depends on two factors: 1) level of spatial correlation in sensor reading 2) the threshold  $\tau$ . We assume that the distance  $D_{ij}$  is a single factor that determines data correlation.  $P_c$  is proportional to threshold  $\tau$ , but inversely proportional to the distance  $D_{ij}$ . Based on this argument, we generate a mathematical model of Equation (1) in a two dimensional space as follows:

$$P_c = f(D_{ij}, \tau) = \frac{c}{1 + \frac{\log(D_{ij})}{\tau}} \quad (4)$$

where  $\sum P_c = 1$  and,  
 $D_{ij} = |x_i - x_j| + |y_i - y_j|$  and,  
 $\tau$  is threshold and  $c$  is constant.

In theory,  $\log(D_{ij})$  can impact  $P_c$ . However, in our following analysis of  $N_c$ ,  $\log(D_{ij})$  is treated as a constant  $c_1$  because our mathematical model in Fig. 2(a) is based on the level of correlation (which is a constant corresponding to the correlation coefficient of 7H) observed in temperature data

from the Great Duck Island [13]. Thus, by combining (3) and (4), the expected number of clusters is as follows :

$$\begin{aligned} N_c &= N - (N-1)P_c = N - \frac{c_2}{1 + \frac{c_1}{\tau}}(N-1) \\ &= N - \frac{c_2\tau}{c_1 + \tau}(N-1), \end{aligned} \quad (5)$$

where  $c_1$  and  $c_2$  are constants

Thus, the total number of transmissions for CAG with spatially correlated data is:

$$\begin{aligned} N_{bc} &= N_q + N_c = N + N - (N-1)P_c \\ &= 2N - \frac{c_2\tau}{c_1 + \tau}(N-1), \end{aligned} \quad (6)$$

where  $c_1$  and  $c_2$  are constants

Therefore, the communication overheads of TAG and CAG in terms of number of transmissions are  $2N$  and  $2N - (N-1)P_c$  respectively with correlated data.

2) *Accuracy of result with CAG:* In this Section, we compute  $P_{error}$  which is the probability that the *relative error* is greater than the user-provided error-tolerance threshold  $\tau$ , where relative error is defined by  $E_r = \frac{|\hat{x} - \bar{x}|}{\bar{x}}$  in which  $x$  is the aggregated sensor readings. That is, how much an approximate result computed by CAG (i.e.,  $\hat{x}$ ) is different from the correct result computed by TAG (i.e.,  $\bar{x}$ ).

$$\begin{aligned} P_{error} &= P_{\bar{x}, \hat{x}}\left(\frac{|\hat{x} - \bar{x}|}{\bar{x}} > \tau\right) = P_{\bar{x}, \hat{x}}(|\hat{x} - \bar{x}| > \bar{x}\tau) \\ &= 1 - P_{\bar{x}, \hat{x}}(|\hat{x} - \bar{x}| \leq \bar{x}\tau) \end{aligned} \quad (7)$$

$$\text{where } \bar{x} = \frac{\sum_{i=1}^N v(n_i)}{N}, \hat{x} = \frac{\sum_{i=1}^{N_p} v_p(n_i)}{N_p},$$

and  $v_p(n_i)$  is the value of participating nodes.

Using Equation (7), we can deduce the following theorem for the error bound of CAG.

*Theorem 2.1: (Precision of CAG)* Relative error in the result obtained from the CAG algorithm is guaranteed to be within  $\tau$  only if data is correlated such that  $N \gg k$ , where  $N$  is the total number of sensor nodes and  $k$  is the number of clusters ( $k = N_p$  which is the number of participating nodes).

*Proof:* relative error  $E_r$  is,

$$\begin{aligned} E_r &= \frac{|\hat{x} - \bar{x}|}{\bar{x}} = \frac{N\bar{x} - (N\bar{x} \pm (N-k)\tau)}{N\bar{x}} \\ &= \frac{\pm(N-k)\tau}{N\bar{x}} \\ &= \frac{\tau}{\bar{x}}, \text{ if } N \gg k \\ &= \frac{\tau}{\bar{x}} < \tau, \text{ if } \bar{x} > 1 \end{aligned}$$

Therefore,  $E_r < \tau$  implies  $P_{\bar{x}, \hat{x}}\left(\frac{|\hat{x} - \bar{x}|}{\bar{x}} > \tau\right) = 0$ , which means  $P_{error} = 0$ . That is, if  $N \gg k$  and  $(\bar{x}) > 1$ , the error in the approximate result from CAG is bounded by  $\tau$ . ■

Note that the above theorem requires  $N \gg k$ , i.e., the total number of sensor nodes is far greater than the number of clusterheads. This requirement is satisfied when the entire data set is spatially correlated enough such that  $(N-k) \cong N$ . It is non trivial to determine the magnitude of  $N$  and  $k$  at which this property holds, and this is a topic of our future work. Later

in Section III, we show that ecological and real sensor data is usually strongly correlated ( $\geq 7H$ ) by using simulations. Due to this high correlation of sensor data, precision with even large  $\tau$  values is shown to be close to the exact result.

We generalize the accuracy of result with random and correlated data. For the same  $\tau$  and  $N$ , as  $N_c(Random) \gg N_c(Correlated)$ , so  $N_p(Random) \gg N_p(Correlated)$  as shown in II-B.1. As the density of nodes increases, the number of clusters with random data increases much faster than the number of clusters formed with correlated data, but  $P_{error}(Random) > P_{error}(Correlated)$  because the correlated data reduces the relative error as shown in *Theorem 2.1*.

To compute the closed-form expression of  $P_{error}$ ,  $P_{\bar{x}, \hat{x}}(|\hat{x} - \bar{x}| \leq \bar{x}\tau)$  in Equation (7) can be described in the same way as in Equation (1). We assume that  $x$  is normalized to  $[0, 1]$ . Thus, by variable replacement,

$$\begin{aligned} P' &= P_{\bar{x}, \hat{x}}(|\hat{x} - \bar{x}| \leq \bar{x}\tau) \\ &= \begin{cases} 2\tau\bar{x} & \text{if } \bar{x} \leq \frac{1}{1+\tau}, \\ 1 - (1-\tau)\bar{x} & \text{if } \bar{x} > \frac{1}{1+\tau}. \end{cases} \end{aligned} \quad (8)$$

Hence,

$$\begin{aligned} P_{error} &= 1 - P' \\ &= \begin{cases} 1 - 2\tau\bar{x} & \text{if } \bar{x} \leq \frac{1}{1+\tau}, \\ (1-\tau)\bar{x} & \text{if } \bar{x} > \frac{1}{1+\tau}. \end{cases} \end{aligned} \quad (9)$$

Fig. 2(b) shows the following mathematical model for the relative error ( $E_r$ ) deduced from the simulation model.

$$\begin{aligned} E_r &= f(\tau) = \log(\tau + 2) - N_c/N \\ &= \log(\tau + 2) - (N - \frac{\tau}{\tau + 1}(N-1))/N \end{aligned} \quad (10)$$

Therefore, the relative error depends on two factors: 1) the level of correlation in data 2) the threshold  $\tau$ .  $E_r$  is proportional to  $\log(\tau)$  and  $-N_c/N$ , where  $N_c$  can be a good indicator of the level of correlation.

### III. PERFORMANCE EVALUATION

In this Section, we performed several experiments to measure 1) the efficiency and precision tradeoff 2) the effect of density on efficiency, and 3) the effect of link reliability on precision.

#### A. Data Sets

For our simulation study, we used three different data sets: synthetic data using a statistical model, synthetic data based on the ecological fractal model, and real sensor data gathered from Great Duck Island. Because more than 99.9% of raw temperature data from Great Duck Island is distributed between 3500 and 6500 raw ADC values, we generated our synthetic data values in the same range (but different levels of correlation) assuming we had a general knowledge of the data distribution before deploying sensors.

1) *Synthetic data from the statistical model:* Sensor data is generated using the method suggested in [8] for a  $250m \times 250m$  two-dimensional grid. Five data sets with different degrees of correlation are generated with parameters  $\alpha = 1/2^i$ ,  $\beta$ , and

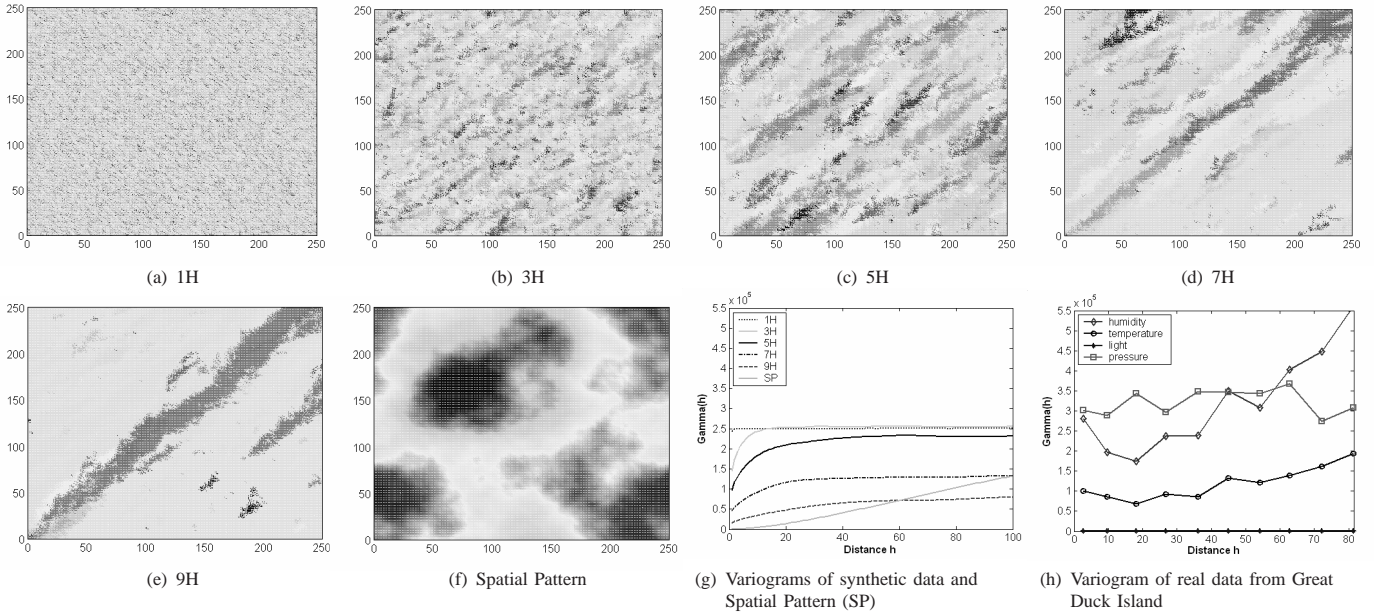


Fig. 3. (a-e) Synthetic data with different correlation ratio, (f) ecological spatial pattern, and (g-h) variograms as described in the caption

$H = 1, 3, 5, 7,$  and  $9$ . Correlation coefficient  $H$  determines the level of correlation; an  $H$  of  $1$  generates data with almost no spatial correlation (similar to *i.i.d.* random), and a larger  $H$  results in a higher spatial correlation. Figures 3(a) to 3(e) show the raw data graphically in  $250m \times 250m$  depending on  $H$ . The variograms of all five data sets are presented in Fig. 3(g). The variogram, also called semivariance, is the most common way to characterize the correlation between pairs of points separated by a spatial distance [1]. In probabilistic notation, the variogram is defined as follows:  $\gamma(h) = \frac{1}{2}E[(X(p) - X(p+h))^2]$  for all possible locations  $p$ , where  $X(p)$  and  $X(p+h)$  are the values at the head and tail of each pair of points with the distance  $h$ .

2) *Synthetic data from the ecology model:* We use the model provided in [10] to generate spatially correlated data with ecological (environmental) patterns (Fig. 3(f)). Even though this data is synthetic, it contains realistic spatial patterns with known spatial properties. Fig. 3(g) includes the variogram of this pattern. This spatial pattern presents the fractal characteristic of the environment with a high correlation level between  $7H$  and  $9H$ .

3) *Real sensor data from Great Duck Island:* Four kinds of modality (humidity, temperature, light, and pressure) measured on Great Duck Island [13] constitute this data set. Different modalities are in different units, but we used raw values in all cases. Variograms using real sensor data from Fig. 3(h) and synthetic data from Fig. 3(g) show similar magnitude and pattern (with each modality corresponding to different level of correlation in (g)). Thus, the synthetic data generated is a good estimate of the real sensor data. As these sensor nodes are not deployed in a grid, distances are subdivided into a number of intervals called *lags* to simplify variogram computation [1].

## B. Simulation Setup and Metrics

We used the Nido simulator [11] for our simulation study. We randomly placed 375 nodes in a  $250m \times 250m$  grid which results in at least a 5-hop topology with each node having an average of 17 neighbors. We used the loss profile from [21] to assign reliabilities to links between nodes. We generated 10 different topologies with this configuration and averaged results over 30 runs for each topology. We configured the bridge nodes to participate in aggregation. We implemented Average, Count, Sum, and Standard Deviation aggregation operators but only present the results for Average in this paper. We chose  $\tau = 0, 0.5, 1, 2, 4,$  and  $10$  because the maximum variation (i.e., the difference between mean and maximum (or minimum)) in data is 25.8% in more than 99.9% of the entire synthetic data set. We also generated topologies with 100% reliable communication, and compared results with those from lossy topologies to understand the effect of packet loss on precision. We ran simulation using two other densities: sparse (9 neighbors/node) and dense (25 neighbors/node). Each node at position  $(x,y)$  uses the value from the corresponding position in the synthetic or empirical data sets.

The primary metric used for evaluation is the percent of the *reduced number of transmissions* calculated as  $\frac{nTX(TAG) - nTX(CAG)}{nTX(TAG)} \times 100$ , where  $nTX$  is the number of transmissions. The number of packets transmitted excludes query packets because it is the same in both TAG and CAG (regardless of  $\tau$ ). Transmission cost is a good estimate of energy cost in WSN because radio transmissions consume far more energy than any other operation in a node. Another metric is the *relative error* of the result with a given  $\tau$  calculated as  $\frac{|EstimateResult - CorrectResult|}{CorrectResult} \times 100$ . Finally, the reduced number of transmissions was compared in 3 different densities.

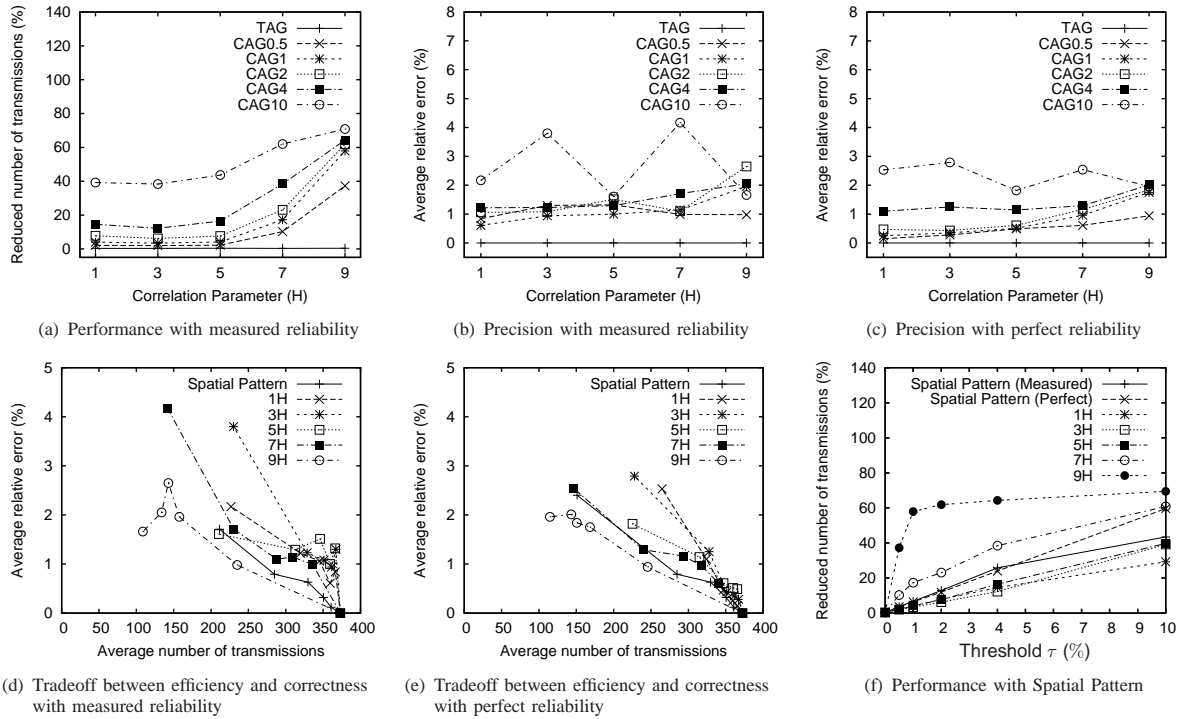


Fig. 4. Simulation result on performance and precision with synthetic data. CAG with  $\tau = n\%$  is termed *CAG $n$*

### C. Results on Efficiency and Precision

As shown in Fig. 4(a), CAG performs fewer transmissions, as more error is allowed in the result by increasing  $\tau$ . With highly correlated data (9H), CAG with  $\tau$  of 0.5% and 1% resulted in 37.3% and 57.9% savings, in communication cost over TAG. Fixing  $\tau$ , with increasing data correlation, we observed reduction in communication overhead. With a  $\tau$  of 4% and data generated with 7H and 9H, we observed CAG has 38.4% and 64.3% reduction in the number of transmissions over TAG. Results with lossless topologies are similar and are omitted due to space constraint.

Fig. 4(b) shows the correctness of the result returned by CAG. With lossy topology (unreliable links), errors increase non-monotonically with  $\tau$ . With lossless topologies (100% link reliability), the result was as expected as shown in Fig. 4(c), since a larger  $\tau$  always results in a larger error. With both topologies, an increase in H was accompanied by an increase in error due to the high disparity between clusterhead and non-clusterhead sensor readings. However, with lossy links, a  $\tau$  of 4 and an H of 7 result in a small relative error of 1.7%. With all  $\tau$  and H values, relative error is always bounded by  $\tau$  except for a few cases with  $H = 9$ , where the error differs by less than 1%. This discrepancy can be attributed to the cluster size oblivious aggregation in CAG. In some simulation runs, we found a large number of clusterheads in the narrow diagonal band of Fig. 3(e). Aggregate from those clusters, when combined (averaged) with aggregate from relatively few clusters from elsewhere, results in an error biased towards that of the narrow band. To address this problem, the next version of CAG takes into account the cluster size while computing aggregates.

Fig. 4(d) and Fig. 4(e) show the communication and error tradeoff with lossy and lossless topologies. We found that the results using spatial pattern from ecology data described in Section III-A.2 are similar to those using synthetic data of 7H and 9H as described in Section III-A.1. This may indicate that the environmental phenomena is highly correlated, consistently staying between 7H and 9H. If we only consider the reduced number of transmissions metric, spatial pattern shows a similar correlation property with 5H and 7H as presented in Fig. 4(f).

Fig. 5(a) and Fig. 5(b) present performance and relative error using the empirical data from Great Duck Island to verify the existence of different levels of spatial correlation in real sensor readings. While the relative errors for pressure and temperature are almost always bounded by the threshold  $\tau$ , the relative errors for light and humidity are generally greater than  $\tau$ . Pressure reading maximally benefits from using CAG more than any other modalities, both in terms of performance and relative error because it is the most strongly correlated environmental attribute in all these data set.

Fig. 5(c) presents the reduced number of transmissions for different densities when  $\tau$  is fixed at 0.5%. With a denser node deployment, CAG saves more energy by exploiting the increased correlation in readings from closeby sensors. Sparse deployment results in weaker data correlation, which increases the energy overhead for CAG.

In short, the following characteristics were observed between correlation level, threshold, and performance and precision of CAG. As data becomes more correlated and the user-given threshold larger, fewer transmissions were observed. When  $\tau < 10\%$ , with higher data correlation and larger threshold, higher relative error was observed. When  $\tau > =$

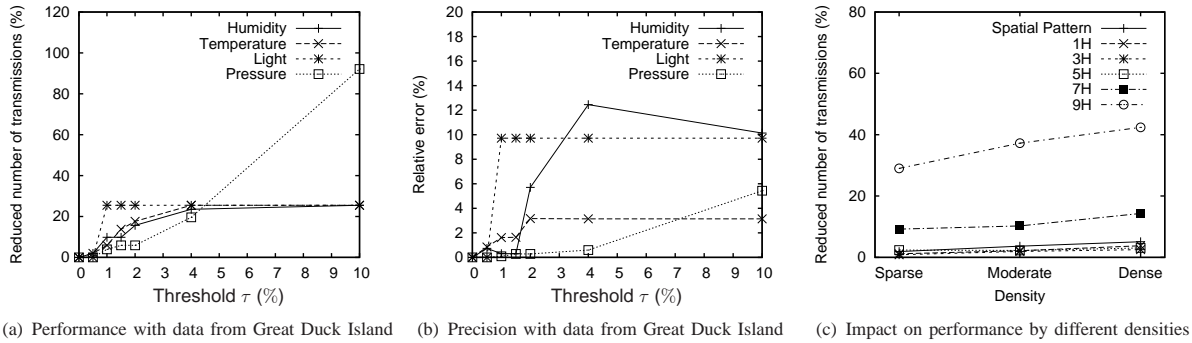


Fig. 5. Simulation result on performance and precision with real sensor data as well as performance on different densities

10%, lower relative error was observed (always bounded by the threshold) as data becomes more correlated. (We also conducted simulations with  $\tau = 15\%$ ,  $20\%$ , and  $40\%$ , but we omit those results due to the space constraint.) We found that the CAG algorithm can maximally take advantage of the highly correlated sensor data.

With CAG, each response from a clusterhead represents the readings of all the nodes within a cluster. Thus, even a single packet loss can seriously affect the accuracy of the final result. Although lossy communication can have a non-linear impact on precision, the magnitude of the increased error due to packet losses is marginal, e.g., less than 2% with synthetic data. The relative error with perfect communication is always lower than the user specified error threshold.

Results from Section II assert that the relative error is guaranteed to be within the user-provided threshold  $\tau$ , when the data shows a certain level of spatial correlation. We observed that the relative error is bounded by  $\tau$  for synthetic, ecological, and real sensor data except humidity and light readings. Although packet losses in the network are expected to have a large negative impact on accuracy, our study shows that this is not the case. This is because the existence of high correlation in sensor reading results in dampening the impact from packet loss. With retransmissions and the highly reliable radios of the Mica2 or MicaZ motes, the errors will be even smaller.

Although the overhead and complexity of forming clusters and implementing threshold-based functionality are drawbacks of CAG, those are negligible because cluster is built during query forwarding; a step common to any query/response protocol in wireless sensor network.

#### IV. CONCLUSION AND FUTURE WORK

In this paper, we presented our novel CAG algorithm that leverages spatial correlation to improve efficiency in WSN. A systematic study of efficiency and precision tradeoff was performed. With highly correlated sensor data, we found that CAG can save the communication overhead by as much as 70.9% over TAG while incurring a modest 1.7% error. In the near future, we plan to validate our simulation results by capturing the real spatial sensor data using the Berkeley mote platform. We also plan to work on a more sophisticated model and analysis that deal with more complex scenarios.

#### REFERENCES

- [1] Mark R.T. Dale, Philip Dixon, Marie-Josée Fortin, Pierre Legendre, Donald E. Myers, Michael S. Rosenbreg, "Conceptual and mathematical relationships among methods for spatial analysis", *Ecography* Vol.25 no.5, October 2002
- [2] Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, Wei Hong, "Model-Driven Data Acquisition in Sensor Networks", VLDB, August - September 2004
- [3] Deepak Ganesan, Ben Greenstein, Denis Perelyubskiy, Deborah Estrin, John Heidemann, "An Evaluation of Multi-resolution Storage for Sensor Networks", *SenSys*, November 2003
- [4] Samir Goel, Tomasz Imielinski, "Prediction-based Monitoring in Sensor Networks: Taking Lessons from MPEG", *ACM CCR*, 2001
- [5] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor networks", *HICSS*, January 2000
- [6] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Mobicom*, August 2000
- [7] Ankur Jain, Edward Y. Chang, Yuan-Fang Wang, "Adaptive Stream Resource Management Using Kalman Filters", *SIGMOD*, June 2004
- [8] Apoorva Jindal, Konstantinos Psounis, "Modeling spatially-correlated sensor network data", *SECON*, October 2004
- [9] Sukun Kim, David Culler, James Demmel, "Structural Health Monitoring Using Wireless Sensor Networks", <http://www.eecs.berkeley.edu/~binetude/course/cs294.1/paper.pdf>
- [10] Jack Lennon, "Red-shifts and red herrings in geographical ecology", *Ecography* Vol.23 no.1, February 2000
- [11] Philip Levis, Nelson Lee, Matt Welsh, David Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", *SenSys*, November 2003
- [12] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, "TAG: Tiny AGgregation service for ad-hoc sensor networks", *OSDI*, December 2002
- [13] Alan Mainwaring, Robert Szewczyk, Joseph Polastre, John Anderson, "Habitat Monitoring on Great Duck Island", <http://www.greatduckisland.net/>
- [14] Arati Manjeshwar, Dharma P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks", *IPDPS*, April 2001
- [15] Arati Manjeshwar, Dharma P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", *IPDPS*, April 2002
- [16] Chris Olston, Jing Jiang, and Jennifer Widom, "Adaptive filters for continuous queries over distributed data streams", *SIGMOD*, June 2003
- [17] Sundeep Pattem, Bhaskar Krishnamachari, Ramesh Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks", *IPSN*, April 2004
- [18] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, Panos K. Chrysanthis, "TiNA: A Scheme for Temporal Coherency-Aware in-Network Aggregation", *MobiDe*, September 2003
- [19] Alec Woo, Samuel R. Madden, Ramesh Govindan, "Networking Support for Query Processing in Sensor Networks", *CACM*, June 2004
- [20] Yong Yao, Johannes Gehrke, "Query Processing for Sensor Networks", *CIDR*, January 2003
- [21] Jerry Zhao, Ramesh Govindan, "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks", *SenSys*, November 2003