Collaborative Data Gathering in Wireless Sensor Networks using Measurement Co-Occurrence

Konstantinos Kalpakis Shilang Tang
Department of Computer Science & Electrical Engineering
University of Maryland Baltimore County
Email: {kalpakis, stang2}@csee.umbc.edu

Abstract

Data gathering is a basic activity of many wireless sensor network applications. We propose a novel collaborative data gathering approach utilizing data co–occurrence, which is different from data correlation. Our approach offers a trade–off between communication costs of data gathering versus errors at estimating the sensor measurements at the base station, by having sensors with co–occurring measurements alternate in transmitting such measurements to the base station, and having the base station make inferences about sensor measurements utilizing only the transmitted data. We describe two effective methods for in-network detecting measurements co-occurrence among sensors, an efficient protocol for scheduling the transmission of measurements, and a simple algorithm for measurement inference. Our simulation results on synthetic and real datasets show a substantial (up to 65%) reduction on the communication costs of data gathering with few number of inference errors ($< 6\%$) at the base station.

1 Introduction

Recent advances in hardware development have led to the creation of wireless ad hoc networks of battery–powered microsensors (WSNs). Because of its unattended operation mode and easy deployment, WSN becomes attractive to many applications such as wildlife tracking, environmental and habitat monitoring, battlefield intelligence, and etc. However, the limited energy of those sensors poses the challenge of using such systems in an energy efficient manner.

We consider the problem of energy efficient data gathering, which is a common activity of many WSN applications. We focus on applications in which each sensor continuously monitors the targets of interests in a field, and the base station is interested in getting every (discrete enumerated) measurement from all the sensors, in order to determine the status of the observing field and make appropriate decisions. Example of such applications can be found in environmental monitoring, quality control in manufacturing assembly lines, agriculture, etc. A simple but energy inefficient method for gathering the measurements is to have each sensor transmitting its every measurements to base station, since often there is redundancy and/or dependency among the sensor measurements.

We propose a new exploitation on data redundancy by a novel collaborative data gathering approach utilizing co–occurrence of measurements, offering a trade-off between communication costs of data gathering versus errors at estimating the sensor measurements at the base station. Intuitively, two sensors have measurements co–occurring if the occurrence sets of the measurements — the set of times at which they are measured — are similar. It is utilized as follows. Using the in-network method we present, sensors discover measurement co-occurrence. Then, sensors with co–occurring measurements collaborate, by informing the base station and taking turns in communicating those measurements to the base station. In addition, each sensor may choose a default measurement, which it will not be transmitting to the base station, upon informing the base station of its choice. Being informed of the measurement co-occurrence relationship and the defaults, the base station infers the measurements of the non-transmitting sensors utilizing only the transmitted measurements.

Let us note that data co-occurrence is different from data correlation. Intuitively, correlation attempts to capture monotonicity trends between sequences (e.g. are both sequences increasing?, etc). Co-occurrence does not provide information about such monotonicity trends; instead, it attempts to quantify the trend that two values tend to occur simultaneously, and is capable of handling discrete enumerated data. It is not hard to find sequences with co–occurring values of high frequency, but having correlation coefficient anywhere in the range $(-1, 1)$. This implies that correlation is not an indicator of co–occurrence. Further, data co–occurrence can appear in both densely and sparsely deployed sensor networks, while data correlation is normally expected in densely deployed sensor networks.

We present two effective methods, namely positional min-wise and random projection, for sensors to in-network detect measurement co-occurrence. Both methods compute small–size signatures of the measurement occurrence sets, and then use the signatures to estimate their resemblance. Computing the signatures and estimating the resemblance are both simple, which makes our methods mindful of the limited energy and computation resources of the sensors. As shown in our simulations, while random projection method performs better, both methods are effective, in terms of signature size and accuracy of resemblance estimation.

On utilizing measurement co-occurrence, we present an efficient protocol for sensors to schedule the transmission of co-occurring measurements. For simplicity, we assume that communication links are lossless. Using our protocol, sensors will determine their measurement transmission schedule dynamically, distributively, and near-immediately. Our protocol is aggressive on reducing transmission of measurements — normally just one of the sensors with co-occurring measurements will transmit, and at the same time it ensures that one of the co-occurring measurements will always be communicated to the base station. In situations where data co-occurrence persist, our simulation results show that our approach can offer substantial communication savings, at the price of few number of inference errors [1] — for synthetic datasets it provides up to 65% savings on the communication costs with no more than 6% errors, and for a real dataset it provides 27% savings and 1.53% inference errors.

1.1 Related work

Broder [1] uses min–wise hashing for identifying near–duplicate documents on the web by estimating their resemblance. Datar et al. [2] present min–wise based algorithms for estimating rarity and similarity on data streams. Agarwal and Trachtenberg [3] use counting Bloom filters [4] for estimating the number of differences between remote sets. Resources in these works are not as limited as we have in wireless sensor networks. Random projections is a powerful dimensionality reduction technique with many applications, since it approximately preserves vector norms under some conditions, see for example [5, 6]. We utilize random projection together with features of 0-1 vectors, on estimating the resemblance of two remote sets in resource constraint sensor networks.

Chou et al. [7] exploit correlations for coding the sensor measurements in order to reduce the total number of bits transmitted during data gathering. Gupta et al. [8], assuming that sensors know the data correlation structure, construct connected correlation–dominating sets to reduce the communication cost of data gathering. Sharaf et al. [9] utilize temporal coherence of sensor data to provide approximate answer to aggregation queries with reduced communication costs, while Yoon and Shahabi [10] utilize spatial correlation of sensor data. Goel et al. [11] use spatio–temporal correlation in sensor data for motion detection, Deshpande et al. [12] incorporate time–varying multivariate Gaussian models of sensor data to answer queries, and transmit sensor data when those models do not answer queries within desired accuracy. Our approach is complementary to them and handles discrete enumerated sensor data. Furthermore, the notion of measurement co-occurrence we utilize is a time–varying non–parametric statistical model of the sensor measurement field, which does not require normality as [12] does.

The rest of the paper is organized as follows. We present the technical details of our approach in section 2, the simulation results with synthetic and real datasets in section 3, and the concluding remarks in section 4.

2 Collaborative data gathering using measurement co-occurrence

2.1 Co-occurrence of measurements

Consider a wireless sensor network with a base station and n sensors, with each sensor having an unique identifier (sid). We refer to the sensor with sid i as the sensor i. The time at which measurements are made is assumed to be a sequence of time intervals of equal length, e.g. the measurement time is discrete. Each sensor has a synchronized clock/counter that measures this discrete measurement time. Hereafter, we refer to measurement time simply as time. We assume that the base station needs to know all the measurements sensors make at each time. A *window* is a contiguous sequence of w times, with the jth window W_j being the interval of $[j * w, (j+1) * w)$, for $j = 0, 1, 2, \ldots$. The relative time of a measurement made at time t in a window $W = [t_0, t_1)$ is $\tilde{t} = t - t_0$. Let U_i be the discrete universe (domain) of the measurements sensor i makes. Let $m_{i,t} \in U_i$ be the measurement sensor i makes at time t. An *element* e is a tuple (i, v), where $v \in U_i$ and i is a sid; for brevity, let $e.value = v$ and $e.sid = i$. Given an element e, we define its occurrence set $\chi_W(e)$ within a window W to be the set of relative times that sensor $e.sid$ makes measurement $e.value$

$$\chi_W(e) = \{\ \tilde{t}\ :\ m_{e.sid,t} = e.value \text{ and } t \in W\ \} \qquad (1)$$

The *resemblance* $r(S_1, S_2)$ of two sets S_1, S_2 is defined as $r(S_1, S_2) = |S_1 \cap S_2| / |S_1 \cup S_2|$. The resemblance of sets takes values between 0 and 1 and is a measure of set similarity.

We say that two elements e_1 and e_2 *co-occur* in window W if the resemblance $r(\chi_W(e_1), \chi_W(e_2))$ is $\geq\ \tau$,

[1] In this work we only consider the number of inference errors. We defer consideration of controlling the magnitude of errors to future work.

where the co-occurrence threshold τ is a constant system parameter $0 \leq \tau \leq 1$. We can show that the probability $\Pr[e_1|e_2] \geq \tau(1+2\tau)/(1+\tau)^2$.

Note that co-occurrence is not a transitive relation, therefore additional care is needed to determine a set of co-occurring elements $\mathcal{L} = \{e_1, e_2, \ldots, e_m\}$. We consider two approaches. In the clique approach, we require that every pair of elements in $\mathcal{L}$ has resemblance $\geq \tau$. In the connected–components (CC) approach, we require that for every pair of elements in $\mathcal{L}$ there exists a chain of elements in $\mathcal{L}$, with adjacent elements having resemblance $\geq \tau$. We experimentally find that the CC approach offers a better communication cost vs. error rate trade–off.

2.2 Estimating co-occurrence of measurements

Having sensors exchange measurement occurrence sets to identify co-occurrence can be unnecessarily expensive. We describe two in-network methods for sensors to detect measurement occurrence at a much smaller communication cost.

First, we present a method for estimating set resemblance that is based on min–wise hashing. Given k random min–wise independent hash functions $h_i : [0, w) \to \mathcal{N}$, $i = 1, 2, \ldots, k$, the *min–wise hash* of a set $S \subseteq [0, w)$ is the set $\alpha(S) = \{\alpha_i(S)\}$, where $\alpha_i(S) = \min(\{h_i(z) \mid z \in S\})$. We define *positional min–wise hash* of set S to be the vector $(\alpha_1(S), \alpha_2(S), \ldots, \alpha_k(S))$. The estimated resemblance of two sets S_1, S_2, using their positional min–wise hashes, is

$$\hat{r}(S_1, S_2) = |\{i \,:\, \alpha_i(S_1) = \alpha_i(S_2)\}|/k \qquad (2)$$

Second, we consider a method for estimating set resemblance based on random projections. Given k random vectors $z_i \in R^w$, $i = 1, 2, \ldots, k$, with entries that are N(0,1) i.i.d. random variables, for any set $S \subseteq [0, w)$, with indicator vector $s \in \{0,1\}^w$, the random projection of S is the projection $\hat{s} = (s^T \cdot z_1, s^T \cdot z_2, \ldots, s^T \cdot z_k)$ of s onto the vectors $\{z_i\}$. Since we can show that the resemblance of two sets $S_1, S_2 \subseteq [0, w)$ is

$$r(S_1, S_2) = \frac{||s_1||^2 + ||s_2||^2 - ||s_1 - s_2||^2}{||s_1||^2 + ||s_2||^2 + ||s_1 - s_2||^2}, \qquad (3)$$

we can estimate $r(S_1, S_2)$ by using the random projections $\hat{s}_1, \hat{s}_2$ instead of s_1, s_2 in the formula above.

We define the (positional min–wise or random projection) *signature* of an element e within window W to be the (positional min–wise hash or random projection) of its occurrence set $\chi_W(e)$. For brevity, whenever it is clear from the context, we simply talk about the signature of an element e, and we denote it with σ_e. The size of σ_e is equal to k, the number of hash functions or projections used to compute it. As shown in our simulation, k is small compared to w. The computation cost of both methods is $\Theta(wk)$. Note that the min-wise hashing can be computed on-line, and for random projection we use the 0-1 indicator vector s, therefore both methods are mindful of the limited computation resources of sensors.

2.3 Exploiting co-occurrence for data gathering

We show a protocol that the sensors and the base station can use to reduce the communication costs of data gathering by exploiting co-occurrences of measurements. The protocol allows sensors to discover co-occurring elements, to collaborate on sharing the load of communicating such co-occurring elements, and it allows the base station to make inferences about the sensor measurements. Here, we assume that the co-occurrences between elements persist for some period of time, much larger than the window size w.

The base station maintains a co-occurrence (symmetric) relation $C : U \times U \to \{0,1\}$, $U = \cup_{i=1}^{n} U_i$, such that $C[e_1, e_2] = 1$ iff it has been notified that e_1 and e_2 co–occur, together with a list of the default values m_{default} it has been notified of [2]. At the end of the time period t, the base station infers $\hat{m}_{i,t}$ for the measurement $m_{i,t}$ of sensor i at t using the algorithm in Figure 1.

```
BaseStationEstimator()
// Base station computes measurement estimates
    at each time $t$
        foreach sensor $i$ do
            $\hat{m}_{i,t} \longleftarrow \mathbf{null}$
        foreach received MeasurementMsg($e$) do
            $\hat{m}_{e.sid,t} \longleftarrow e.value$
            foreach element $e'$ that co-occurs with $e$, i.e. $C[e', e] = 1$ do
                if $\hat{m}_{e'.sid,t}$ is null then
                    $\hat{m}_{e'.sid,t} \longleftarrow e'.value$
        foreach sensor $i$ do
            if $\hat{m}_{i,t}$ is null then
            $\hat{m}_{i,t} \longleftarrow m_{\text{default}}(i)$

ReceiveNewMsg(set $S$)
// Base station receives NewMsg
    foreach $(e_1, e_2) \in S \times S$ do
        $C[e_1, e_2] \longleftarrow 1$

ReceiveQuitMsg(element $e$)
// Base station receives QuitMsg
    foreach element $e' \in U$ do
        $C[e, e'] \longleftarrow 0$
```

Figure 1. Algorithm used by base station to compute $\hat{m}_{i,t}$.

Sensor i maintains for each of its elements e a data structure $\Phi[e]$ that consists of: a list $\Phi[e]$.list of elements that are discovered as co-occurring with e [3], sorted in increasing order of their sids; a list $\Phi[e]$.children of elements that

[2] Co-occurrence relation C can be maintained using any standard data structure for (sparse) undirected graphs. Each sensor can have only one default value.

[3] Note that sensor i cannot have co-occurrence among its own elements. If e is not co-occurring with other element, then $\Phi[e].\text{list} = \{e\}$ and $\Phi[e].\text{state} = \mathit{normal}$.

```
SensorMeasurementLoop()
foreach window $W$ do
    $\chi_W(e) \longleftarrow \emptyset$ for all $e \in U_i$
    foreach time $t$ within $W$ do
        take a measurement $v$ and create element $e = (i, v)$
        append relative time $\tilde{t}$ to $\chi_W(e)$
        if $IsOnDuty(i, e, \tilde{t})$ then
            send MeasurementMsg($e$) to base station
    // end of window
    if $\tilde{t} = w - 1$ then
        $TSS_i \longleftarrow \emptyset$
        foreach $e \in U_i$ do
            $\sigma_e \longleftarrow$ signature of $\chi_W(e)$
            append the tuple $(e, \sigma_e)$ to $TSS_i$
        if $\Phi[e].\mathrm{state} \neq waiting$ then
            $\Phi[e].\mathrm{state} \longleftarrow normal$

IsOnDuty(sid $i$, element $e$, relative time $t$)
// Sensor $i$ computes its duty status for element $e$ at relative time $t$
if $\Phi[e].\mathrm{state}$ is $waiting$ or $init$ then
    return $\mathbf{true}$
else
    if $e.value = m_{\mathrm{default}}(i)$ or $\Phi[e].\mathrm{state} = updating$ then
        return $\mathbf{false}$
    else
        let $j$ be the index of the element $e$ in $\Phi[e].\mathrm{list}$
        split the window $[0, w)$ into $|\Phi[e].\mathrm{list}|$ intervals $I_0, I_1, \ldots$
            if $t$ is in interval $I_j$ then
                return $\mathbf{true}$
            else
                return $\mathbf{false}$
```

Figure 2. Algorithm used by sensors to schedule element transmissions.

```
DiscoverCoOccurrences(threshold $\tau$)
// Sensor $i$ discovers co-occuring elements at threshold $\tau$
foreach sensor $j \in Adj_i$ do
    request $TSS_j$ from sensor $j$
    foreach tuple $(e, \sigma_e) \in TSS_i$ do
        foreach tuple $(e', \sigma_{e'}) \in TSS_j$ do
            if $r(\sigma_e, \sigma_{e'}) \geq \tau$ then
                append $e'$ to $\Phi[e].\mathrm{list}$ and to $\Phi[e].\mathrm{children}$
                mark $\Phi[e]$ as changed
                break
foreach $e \in U_i$ do
    if $\Phi[e]$ is marked as changed then
        send NewMsg($\Phi[e].\mathrm{list}$) to the base station
        $\Phi[e].\mathrm{state} \longleftarrow waiting$
        foreach $e' \in \Phi[e].\mathrm{list}$ do
            send UpdateMsg($i, \Phi[e].\mathrm{list}$) to sensor $e'.sid$
upon receiving AckMsg for every UpdateMsg sent do
    $\Phi[e].\mathrm{state} \longleftarrow init$

ReceiveUpdateMsg(sid $j$, set $S$)
// Sensor $i$ receives UpdateMsg from sensor $j$
find the element $e \in U_i \cap S$
$\Phi[e].\mathrm{state} \longleftarrow updating$
append to $\Phi[e].\mathrm{list}$ all the elements in $S$
foreach element $e' \in \Phi[e].\mathrm{children}$ do
    if $e' \notin S$ then
        send UpdateMsg($i, \Phi[e].\mathrm{list}$) to sensor $e'.sid$
upon receiving AckMsg for every UpdateMsg sent do
    send AckMsg to sensor $j$
```

Figure 3. Algorithm used by sensors to discover co-occurrence elements.

are determined to co–occur with e by sensor i itself; and an attribute $\Phi[e].\mathrm{state}$ indicating the status of $\Phi[e]$. The attribute $\Phi[e].\mathrm{state}$ takes values (a) $normal$ if $\Phi[e]$ is up–to–date, (b) $waiting$ if sensor i initiated an update and is waiting for acknowledgment messages, (c) $init$ if sensor i had been the initiator of an update in the current window, or (d) $updating$ if sensor i received an update message UpdateMsg in the current window.

At each time t, sensor i decides whether to transmit to the base station its measurement using the algorithm in Fig 2. In this algorithm, the current window is partitioned into equi–length sub-windows, called *duty–zones*, with each sensor in $\Phi[e].\mathrm{list}$ taking charge of one duty–zone. This makes the scheduling simple and distributive, and enables sensors to join or leave $\Phi[e].\mathrm{list}$ quickly. There are different ways to split a window into duty–zones, for example, using equi–depth approach, the window can be split into duty–zones so that each one has approximately the same number of occurrences of the co-occurring elements. The equi–depth approach tries to distribute the burden of communicating the co-occurring elements equally among the sensors in $\Phi[e].\mathrm{list}$. Such an approach may lead to longer network lifetimes, provided the overhead in computing equi–depth duty zones is small. Also it may be attractive for sensors with more residual energy in $\Phi[e].\mathrm{list}$ to take charge of multiple duty–zones. We defer these considerations to future work.

Sensor i maintains a set TSS_i with its elements and their signatures in the previous window, which it updates at the end of the current window.

Sensor i may initiate the discovery of elements that may co-occur with one of its elements e at any time, by requesting the signatures of elements from neighboring sensors, using the connected–components (see Fig. 3) or clique approach. Sensor i requests the set TSS_j of signatures of elements from sensors $j \in Adj_i$, and if it finds an element e' that co-occurs with e, it updates $\Phi[e]$ by adding e' to $\Phi[e].\mathrm{list}$ and $\Phi[e].\mathrm{children}$, and then updates the base station and all the sensors with an element in $\Phi[e].\mathrm{list}$. Whenever a sensor receives such an update, it further updates all its children not already updated.

Sensor i may quit having e in the $\Phi[e].\mathrm{list}$ at any time [4]. In such a case, sensor i removes e from $\Phi[e].\mathrm{list}$, chooses a sensor j in $\Phi[e].\mathrm{list}$ to act as a quit coordinator for updating the remaining sensors with elements in $\Phi[e].\mathrm{list}$. The quit coordinator sensor j “adopts” e’s children $\Phi[e].\mathrm{children}$, and it tells all sensors with an element in $\Phi[e].\mathrm{list}$ to remove e from their co-occurrence lists. See Figure 4 for details.

Because the measurement co-occurrence is estimated approximately, the base station may make *inference errors* $\hat{m}_{i,t} \neq m_{i,t}$ when sensor i is *off-duty* at time t. Here we briefly analyze when inference errors may happen, and our focus in this paper is the number rather than the magnitude of inference errors. When sensor i has default element $m_{\mathrm{default}}(i)$ but not any co-occurring element, or does not have $m_{\mathrm{default}}(i)$ but only one co-occurring element, base station will always be able to infer $\hat{m}_{i,t}$ correctly as $m_{i,t}$, i.e. it makes no inference error. When sensor i has multiple co-occurring elements and is off-duty for more than

[4] For example, the base station may request sensor i to quit or re-establish having e in the $\Phi[e].\mathrm{list}$ when it suspects high number or costly errors on e. Furthermore, if sensor i has an element e with high frequency but short $\Phi[e]$.list, it may decide to quit e from $\Phi[e].\mathrm{list}$ and choose e as its default $m_{\mathrm{default}}(i)$.

```
Quit(element $e$)
// Sensor $i$ quits co-occurrences for its element $e$
    send QuitMsg($e$) to base station
    choose a sensor $j$ as quit coordinator among the sensors with elements in $\Phi[e]\texttt{.list}$
    send QuitCoordinateMsg($e$, $\Phi[e]\texttt{.children}$) to sensor $j$
    set $\Phi[e]\texttt{.list} \longleftarrow \{e\}$, $\Phi[e]\texttt{.state} \longleftarrow normal$, and $\Phi[e]\texttt{.children} \longleftarrow \textbf{null}$

ReceiveQuitCoordinateMsg(element $e$, set $S$)
// Sensor $i$ receives QuitCoordinateMsg
    find element $e' U_i$ that co-occurs with $e$, e.g. $e \in \Phi[e']\texttt{.list}$
    remove $e$ from both $\Phi[e']\texttt{.list}$ and $\Phi[e']\texttt{.children}$
    append to $\Phi[e']\texttt{.children}$ all the elements in $S$
    $\Phi[e']\texttt{.state} \longleftarrow waiting$
    foreach $e'' \in \Phi[e']\texttt{.list}$ do
        send QuitMsg($i$, $e$, $\Phi[e']\texttt{.list}$) to sensor $e''\texttt{.}sid$
        upon receiving AckMsg for each QuitMsg sent do
        $\Phi[e']\texttt{.state} \longleftarrow init$

ReceiveQuitMsg(sid $j$, element $e$, set $S$)
// Sensor $i$ receives QuitMsg from sensor $j$
    find the element $e' \in U_i \cap S$
    $\Phi[e']\texttt{.state} \longleftarrow updating$
    remove $e$ from both $\Phi[e']\texttt{.list}$ and $\Phi[e']\texttt{.children}$
    foreach $e'' \in \Phi[e']\texttt{.children} - S$ do
        send QuitMsg($i$, $e$, $S$) to sensor $e''\texttt{.}sid$
    upon receiving AckMsg for every QuitMsg sent do
        send AckMsg to sensor $j$
```

Figure 4. Algorithm used by sensors to quit element from co-occurences.

one of them at time t, even if no sensor has m_{default}, the base station may erroneously infer $\hat{m}_{i,t} \neq m_{i,t}$, because it may have to guess which one of the off-duty co-occurring elements was measured by sensor i at time t. Similarly when sensors have m_{default}, even if sensor i has just one co-occurring element e, the base station may make errors in $\hat{m}_{i,t}$, as it may not correctly determine if e or $m_{\text{default}}(i)$ was measured by sensor i at time t.

3 Experimental evaluation

We use both synthetic datasets and real sensor measurements in our experiments. For synthetic datasets, we used a uniform distribution model for generating sensor measurements. For the real sensor datasets, we downloaded from [13] the temperature and humidity measurements of sensors in James Reserve, CA.

To evaluate our co–occurrence detection methods, we use the average resemblance estimation error, and the size k of the signatures σ_e. To evaluate the utilization of co–occurrence on data gathering, we use the inference error rate — number of inference errors over number of total measurements — and the relative reduction of the communication costs for data gathering due to the proposed method. Here we only consider the number of inference errors. We are aware of the magnitude of the inference errors could be arbitrarily high and hard to predict. We defer this part to future work. The communication overhead of our approach, due to the discovery and quit algorithms, is proportional to the size of the signatures and the number of signatures exchanged. As long as co-occurrences persist for a few windows, this overhead will be small compared to the savings in the measurement communication costs.

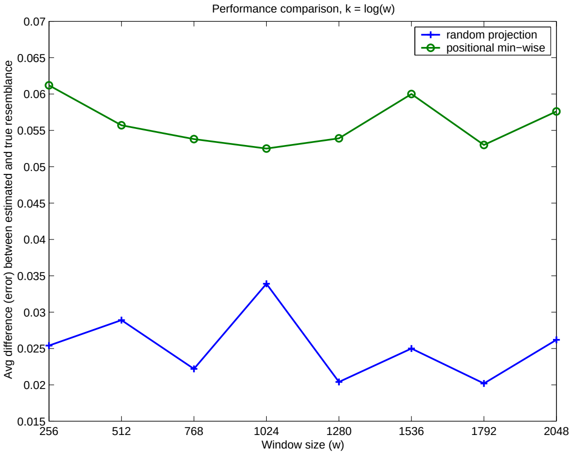

Figure 5. Performance of positional min-wise and random projection methods, $k = \log w$.

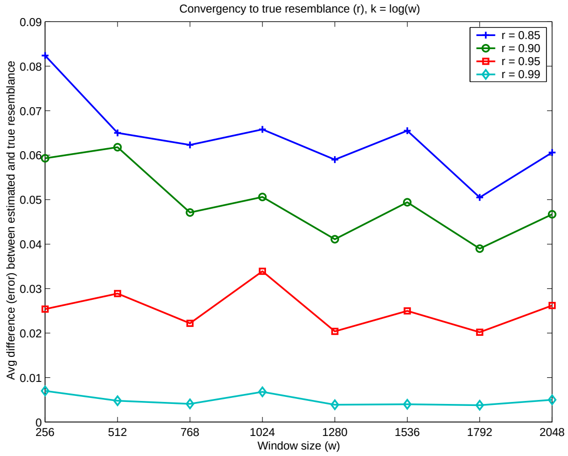

Figure 6. Convergence of random projection method.

3.1 Experimental results – synthetic datasets

To evaluate the performance of our in-network data co-occurrence detection methods, we varied the window size w from 256 to 2048, and for each w we generated 100 pairs of occurrence sets with frequency $f = 0.3$, i.e., each set having size $\approx f \times w$, as our datasets. The true resemblance between each pair of sets is ≈ 0.95. Fig 5 shows the performance of both resemblance estimation methods, with $k = \log w$. We see that $\log w$ works well for both, while the random projection method gives more accurate resemblance estimation than the positional min-wise method. Hereafter, we use random projection method with $k = \log w$.

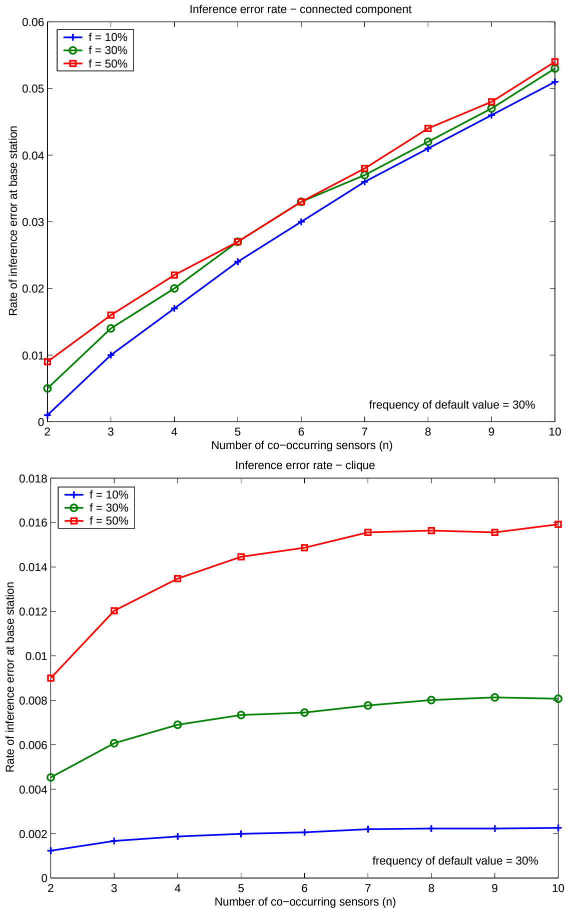

Figure 7. Base station inference error rate using the connected–components and clique approaches.

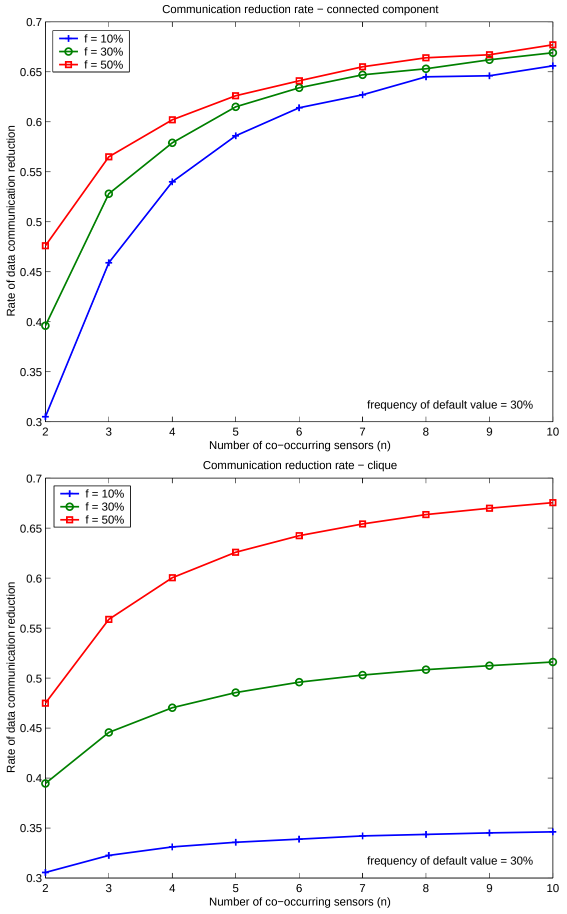

Figure 8. Reduction of communication costs using the connected–components and cliques approaches.

Next, in Fig 6, we examine the estimation behavior of random projection method for sets with different true resemblance. As before, for each window size w from 256 to 2048 we generated 100 pairs of sets with $f = 0.3$ as our datasets. We see that the error in estimating resemblance decreases as as the true resemblance increases. This is important because it allows us to effectively use τ for exploiting the trade–off between number of inference errors and communication costs.

We consider two approaches, the clique and the CC approach, for identifying sets of $n \geq 2$ co-occurring elements. These two approaches affect both the number of inference errors as well as the communication costs. For these experiments, we consider sets of $n = 2, 3, \ldots, 10$ elements with varying frequency f, window size $w = 1024$, true resemblance between pairs of elements 0.95, signature size $k = \log w$, and frequency of the default element chosen by each sensor 30%. We run both approaches for 100 consecutive windows, with the results given in Figs 7 and 8.

Fig 7 gives the base station inference error rate, while Fig 8 shows the reduction of communication costs. We clearly see that the error rate is very low, and the reduction of communication costs is substantial and increases as n increases. Both approaches achieve comparable communication cost savings for the same number of sensors with co-occurring elements. As expected, the inference error rate with the clique approach is lower and increases slower with n as compared with the CC approach. However, this advantage of the clique approach comes with higher communication and computation overheads, since the signatures of every element in $\Phi[e].\mathrm{list}$ needs to be examined before an element e' can join $\Phi[e].\mathrm{list}$. Moreover, the clique approach may result in smaller $\Phi[e].\mathrm{list}$, leading into smaller reduction of communication costs. Consequently, the CC approach should be preferred for most applications.

3.2 Experimental results – real dataset

To evaluate how our method might perform in real sensor networks, we downloaded the temperature and humidity measurements taken, every 5 minutes, by 12 sensors deployed in James Reserve, CA during Aug. 9 and 10, 2005. [5] We use window size $w = 512$, random projection method with the CC approach for identifying sets of co-occurring elements, and $\tau = 0.90$. We used the measurements in the first window for co-occurrence detection and found six groups of co-occurring elements among the temperature measurements: one with 6 elements, one with 3 elements, and four with 2 elements. Each sensor selects as its default element the most frequent temperature measurement that does not co–occur with any other element. It turns out, that following our scheme, during the first window, there were 1663 fewer measurements out of $6144 (= 12 \times 512)$ transmitted to the base station, while the number of inference errors was 94. In other words, our scheme provides a 27.1% reduction of the communication cost for data gathering with an error rate of 1.53%.

4 Conclusion

We describe a novel approach to the data gathering problem in wireless sensor networks that is based on the idea of measurements co-occurrence. We present two in-network methods for sensors to detect co-occurring data, a protocol for sensors to collaborate in transmitting co-occurring measurements to a base station, and a measurement inference algorithm for the base station. Our approach provides a new exploitation on data redundancy for communication cost vs inference errors trade–off.

We experimentally evaluate the proposed approach, and find that it offers substantial savings in the communication costs for the price of few number of errors at the base station. For synthetic datasets it provides up to 65% savings on the communication costs with no more than 6% inference errors, and for a real dataset it provides 27% savings and 1.53% inference errors.

References

[1] A. Z. Broder. Identifying and filtering near-duplicate documents. *CPM 2000, LNCS 1848*, pages 1–10, 2000.

[2] M. Datar and S. Muthukrishnan. Estimating similarity and rarity over data stream windows. In *Proceedings of the 10th European Symposium on Algorithms*, Rome, Italy, Sept. 2002.

[3] S. Agarwal and A. Trachtenberg. Estimating the number of differences between remote sets. In *IEEE Information Theory Workshop (ITW)*, Punta del Este, Uruguay, 2006.

[4] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, June 2000.

[5] D. Achlioptas. Database-friendly random projections: Johnson–lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 26:671–687, 2003.

[6] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk. Universal distributed sensing via random projections. In *ISPN '06: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*. ACM Press, 2006.

[7] J. Chou, D. Petrovic, and K. Ramchandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *Proceedings of the IEEE INFOCOM*, 2003.

[8] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary. Efficient gathering of correlated data in sensor networks. In *MobiHoc'05*, Urbana-Champaign, Illinois, May 2005.

[9] M. Sharaf, J. Beaver, A. Labrinidis, and P. Chrysanthis. Tina: A scheme for temporal coherency-aware in-network aggregation. In *MobiDE'03*, San Diego, CA, USA, September 2003.

[10] S. Yoon and C. Shahabi. Exploiting spatial correlation towards an energy efficient clustered aggregation technique (cag). In *Proceedings of the International Conference on Communications (ICC)*, 2005.

[11] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: taking lessons from mpeg. *SIGCOMM Comput. Commun. Rev.*, 31(5):82–98, 2001.

[12] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the 30th VLDB Conference*, Toronto, 2004.

[13] James reserve data management system. http://cens.jamesreserve.edu/jrcensweb/cmstest/cms_env_data_list.php.

[5] No data were available for other days. Each missing measurement was replaced with a new distinct value. We rounded each measurement x to $round(x)$.