ELSEVIER

Available online at www.sciencedirect.com
ScienceDirect

Computer Communications xxx (2008) xxx–xxx

computer communications

www.elsevier.com/locate/comcom

Collaborative data gathering in wireless sensor networks using measurement co-occurrence

Konstantinos Kalpakis *, Shilang Tang

Computer Science & Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

Received 18 April 2007; accepted 6 January 2008

Abstract

Wireless ad hoc networks of battery-powered microsensors (WSNs) are proliferating rapidly and transforming how information is gathered and processed, and how we affect our environment. The limited energy of those sensors poses the challenge of using such systems in an energy efficient manner to perform various activities. A common activity of many applications of WSNs is that of data gathering: for each time step, gather the measurement from each sensor to a base station. Often there is redundancy and/or dependency among the sensor measurements. How to identify the data redundancy/dependency and utilize them on improving energy efficiency of data gathering has been one of the attractive topics.

We propose using measurement co-occurrence to identify data redundancy and a novel collaborative data gathering approach utilizing co-occurrence that offers a trade-off between the communication cost of data gathering versus errors at estimating the sensor measurements at the base station. A key tenant of our approach is to have sensors with co-occurring measurements alternate in transmitting such co-occurring measurements to the base station, and having the base station make inferences about the sensor measurements utilizing only the data transmitted to it. We present two effective in-network methods for detecting co-occurrence of measurements, as well as a simple and efficient protocol for scheduling the transmission of the sensor measurements to the base station.

We provide experimental results on synthetic and real datasets showing that the proposed system offers substantial (up to 65%) reduction of the communication costs of data gathering with a small number of measurement inference errors (<6%) at the base station.

Keywords: Wireless sensor networks; Data gathering; Set resemblance; Co-occurrence

1. Introduction

Recent advances in hardware developments have led to the creation of wireless sensor networks (WSNs). Such networks are envisioned to consist of low-cost sensor nodes operating in unattended mode, each with some limited computational power and low range wireless communication ability, and generally being battery powered. Because of its unattended operation mode and easy deployment, WSNs become attractive to many applications such as wildlife tracking, environmental and habitat monitoring, battlefield intelligence, and etc. However, the limited energy of their sensors poses the challenge of using such systems in an energy efficient manner (see Fig. 1). Q1

We consider the problem of energy efficient data gathering, which is a basic activity of many WSN applications. We focus on applications in which each sensor continuously monitors a field of interest, and the base station is interested in getting every 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 method for gathering the measurements is to have each sensor transmit its every measurement to the base station. However, this method is energy inefficient since often

* Corresponding author. Tel.: +1 410 455 3143.
E-mail addresses: kalpakis@csee.umbc.edu (K. Kalpakis), stang2@csee.umbc.edu (S. Tang).

0140-3664/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.comcom.2008.01.001

BaseStationEstimator()
// Base station computes measurement estimates
at each time t
  foreach sensor i **do**
    $\hat{m}_{i,t} \longleftarrow$ **null**
  foreach received MeasurementMsg(e) **do**
    $\hat{m}_{e.sid,t} \longleftarrow e.value$
  foreach sensor i with $\hat{m}_{i,t} =$ **null** **do**
    compute the extended candidate measurements $\hat{V}_{i,t}$
    let e be the first element in $\hat{V}_{i,t}$
    $\hat{m}_{i,t} \longleftarrow e.value$

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

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

Fig. 1. The algorithm used by the base station to estimate sensor measurements $\hat{m}_{i,t}$.

there is redundancy and/or dependency among the sensor measurements.

Identifying data redundancies/dependencies and utilizing them in order to provide energy efficient data gathering has been considered by many researchers [10,13,8,6,18]. In this paper, we propose a new idea of using measurements (data) co-occurrence to identify data redundancy together with two methods to estimate it and a novel collaborative data gathering approach utilizing the measurements co-occurrence. Our proposed approach offers a trade-off between communication costs of the data gathering versus number of estimation errors of the sensor measurements at the base station. Intuitively, two measurements co-occur if the set of times at which they are measured are similar. We utilize co-occurrence as follows. Sensors identify co-occurring measurements by using the in-network method we present, which relies on estimating the approximate resemblance of the measurement occurrence sets. Then, sensors with co-occurring measurements collaborate, by informing the base station and then taking turns in communicating those measurements to the base station. In addition, each sensor may choose a default measurement, which it does not transmit upon informing the base station of its choice. Being informed of the measurement co-occurrence relationship and the sensor defaults, the base station infers the measurements of the non-transmitting sensors utilizing only the transmitted measurements.

Data co-occurrence is different from data correlation, which is normally expected in densely deployed sensor networks. Data correlation has been exploited to reduce the communication costs for gathering measurements to the base station [13,8,6,18], or for in-network processing of aggregation queries [10,13,19,12,16]. Intuitively, correlation attempts to capture monotonicity trends (e.g. linear dependencies) between sequences. Co-occurrence does not provide information about such monotonicity trends; instead, it attempts to quantify the trend that two values tend to occur simultaneously (e.g. non-linear dependencies), and is capable of handling discrete enumerated data. We can find sequences with co-occurring values of high frequency, and with arbitrary correlation coefficient, which implies that correlation is not an indicator of co-occurrence. Further, data co-occurrence can appear in both densely and sparsely deployed sensor networks.

We present two in-network methods, namely positional min-wise and random projection, for sensors to detect measurement co-occurrence. Both methods compute small-size signatures of measurement occurrence sets, and then use these signatures to estimate the resemblance of the measurement occurrence sets. 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 experiments, while the random projection method performs better, both methods are effective, in terms of signature size and accuracy of resemblance estimation.

In order to utilize measurements co-occurrence, we present an efficient protocol for sensors to coordinate 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. Our experimental results show that our approach offers substantial communication savings, at the price of a small number of inference errors[1] – 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.

The rest of the paper is organized as follows. In Section 2 we discuss in details of measurement co-occurrence and present our methods for estimating co-occurrence. In Section 3 we describe our collaborative data gathering protocol exploiting measurement co-occurrence on reducing data communication costs. In Section 4 we present the results of our experimental evaluation with synthetic and real data. Related work is discussed in Section 5, and conclusions are given in Section 6.

2. Estimating co-occurrence of sensor measurements

2.1. Measurement co-occurrence

Consider a wireless sensor network with a base station and n sensors. Each sensor has a unique identifier (sid).

[1] Our main focus in this paper is the number rather than the magnitude of inference errors.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

We refer to the sensor with sid i as the sensor i. Sensors take measurements of their environment at each time, while the base station needs to know all the measurements sensors make at each time. The time at which measurements are taken is assumed to be discrete. Each sensor has a clock indicating the measurement time. Sensor clocks are locally synchronized (within the neighborhood Adj_i of each sensor i). Hereafter, we refer to measurement time simply as time. A *window* is a contiguous sequence of w times, with the jth window being $W_j = [j * w, (j+1) * w)$, $j \geqslant 0$. The relative time of a measurement made at time t within a window $W = [t_0, t_0 + w)$ 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 $e.sid = i$. The occurrence set $\chi_W(e)$ of an element e for a window W is the set of relative times that sensor $e.sid$ makes measurement $e.value$ within the window W,

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

The *resemblance* $r(S_1, S_2)$ of any two sets S_1, S_2 is defined as

$$r(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \quad (2)$$

Set resemblance takes values between 0 and 1 and is a measure of set similarity, e.g. if $S_1 = S_2$ then $r(S_1, S_2) = 1$ and if $S_1 \cap S_2 = \emptyset$ then $r(S_1, S_2) = 0$.

We are interesting in determining the degree that elements tend to occur (be measured) at the same or almost the same times. We say that two elements e_1 and e_2 *co-occur* in a window W if the resemblance $r(\chi_W(e_1), \chi_W(e_2))$ of their occurrence sets is $\geqslant \tau$, where τ is the co-occurrence threshold, a system parameter between, $0 < \tau \leqslant 1$. Observe that an element of a sensor i can co-occur with at most $\lfloor 1/\tau \rfloor$ elements from sensor $j \neq i$. Moreover, note that co-occurrence is not a transitive relation, i.e. since $r(S_1, S_2) \geqslant \tau$ and $r(S_2, S_3) \geqslant \tau$ does not always imply $r(S_1, S_3) \geqslant \tau$, for any sets S_1, S_2, S_3.[2] Therefore, additional care is needed when using the resemblance of occurrence sets to determine whether a group of three or more elements co-occur.

We consider two different approaches on determining co-occurrence for a set of elements $\mathcal{L}$ at a threshold τ. In the clique approach, each pair of elements in $\mathcal{L}$ is required to co-occur at threshold τ. 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 co-occurring at threshold τ. We experiment with both approaches, and we find that the connected-components approach presents a better trade-off between communication costs vs. error rate.

An element e can be thought of as the event of sensor $e.sid$ measuring value $e.value$. Consider two elements e_1 and e_2. Since the conditional probability of e_2 given e_1 is

$$Pr[e_2|e_1] = \lim_{|W| \to \infty} \frac{|\chi_W(e_1) \cap \chi_W(e_2)|}{|\chi_W(e_1)|}, \quad (3)$$

and the probability of e_1 is

$$Pr[e_1] = \lim_{|W| \to \infty} \frac{|\chi_W(e_1)|}{|W|}, \quad (4)$$

using Lemma 1 in Appendix A, we find that a lower bound τ on the resemblance of the ocurrence sets of e_1 and e_2, in the limit, implies a lower bound of $\tau(1+2\tau)/(1+\tau)^2$ on the conditional probabilities $Pr[e_2 \mid e_1]$ and $Pr[e_1 \mid e_2]$.

Measurement correlation has been used as a way to reduce communication costs in wireless sensor networks, while, to the best of our knowledge, this is the first time that co-occurrence is proposed for that purpose. Correlation and co-occurrence are generally different concepts. Intuitively, correlation attempts to capture monotonicity trends (linear dependencies) between numerical sequences (are both increasing/decreasing? is increasing and the other decreasing? 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 (non-linear dependencies). There are sequences that contain co-occurring elements with large occurrence sets, and which have arbitrary correlation coefficients (see Appenidx B for such example sequences). Therefore, the correlation coefficient is not an indicator of co-occurrence.

2.2. Estimating the resemblance of occurrence sets

The naive approach for two sensors to determine whether two elements e_1 and e_2 co-occur is for sensor $e_2.sid$ to compute the resemblance of the occurrence sets of e_1, e_2 after obtaining the occurrence set $\chi(e_1)$ from sensor $e_1.sid$. The communication cost of this approach can be unnecessarily high. We present two methods for sensors to approximately compute the resemblance of element occurrence sets with smaller communication cost.

2.2.1. Positional min-wise hashing

The first method is based on min-wise hashing. Min-wise hashing has been used before to estimate resemblance of sets. Consider 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

$$\alpha(S) = \{\alpha_i(S) \mid i = 1, 2, \ldots, k\}, \quad (5)$$

where

$$\alpha_i(S) = \min(\{h_i(z) \mid z \in S\}). \quad (6)$$

Given two sets $S_1, S_2 \subseteq [0, w)$, it turns out that

$$Pr[\alpha_i(S_1) = \alpha_i(S_2)] = r(S_1, S_2). \quad (7)$$

The resemblance $r(S_1, S_2)$ of S_1, S_2 can be estimated by

$$\hat{r}(S_1, S_2) = \frac{|\alpha(S_1) \cap \alpha(S_2)|}{k}. \quad (8)$$

Datar and Muthukrishan [9, Lemma 1] show that

[2] On the other hand, it can be shown that if $|S_1| = |S_2| = |S_3|$ then $r(S_1, S_3) \geqslant r(S_1, S_2) + r(S_2, S_3) - 1$.

Theorem 1. ([9]) *For any* $1 > \epsilon, p, \delta > 0$ *and* $k \geqslant 2\epsilon^{-3}p^{-1}\log\delta^{-1}$,

$$\hat{r}(S_1, S_2) = (1 \pm \epsilon) \cdot r(S_1, S_2) + \epsilon p, \tag{9}$$

with probability at least $1 - \delta$.

For example, when $\epsilon = 0.05$, $p = 0.99$, and $\delta = 0.05$, Theorem 1 implies that, in order to estimate resemblance with 95% accuracy and 95% confidence, k (and thus the window size in our case) needs to be >48,416.

We define the *positional min-wise hash* of set S to be the vector $(\alpha_1(S), \alpha_2(S), \ldots, \alpha_k(S))$, and estimate the resemblance of two sets S_1, S_2 using their positional min-wise hashes as

$$\hat{r}(S_1, S_2) = \frac{|\{i : \alpha_i(S_1) = \alpha_i(S_2)\}|}{k}, \tag{10}$$

In our experiments, we find that the positional min-wise hashing approach with $k = 15$ gives an estimated resemblance within 0.05 of the true resemblance for sets $S_1, S_2 \subseteq [0, w)$, where $w \leqslant 2048$. The value $k = 15$ is too small for the standard min-wise hash approach to provide useful resemblance estimation. Hereafter, we use positional min-wise hashing instead of the standard min-wise hashing.

2.2.2. Random projection

The second method we consider for estimating set resemblance is based on random projections. Random projections is a powerful dimensionality reduction technique with many applications, since it approximately preserves vector norms under some conditions [2,14]. Since any set $S \subseteq [0, w)$ has an indicator vector $s \in \{0, 1\}^w$, we refer to a random projection $\hat{s}$ of the vector s as a random projection of the set S.

Consider two sets $S_1, S_2 \subseteq [0, w)$. We can show that $|S_1 \oplus S_2| = \|s_1 - s_2\|^2$ and $|S_1 \cap S_2| = \langle s_1, s_2 \rangle = (\|s_1\|^2 + \|s_2\|^2 - \|s_1 - s_2\|^2)/2$, where $S_1 \oplus S_2 = (S_1 \cup S_2) - (S_1 \cap S_2)$. Furthermore,

$$r(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{|S_1 \cap S_2|}{|S_1 \oplus S_2| + |S_1 \cap S_2|} = \frac{\|s_1\|^2 + \|s_2\|^2 - \|s_1 - s_2\|^2}{\|s_1\|^2 + \|s_2\|^2 + \|s_1 - s_2\|^2}. \tag{11}$$

using the random projections $\hat{s_1}$ and $\hat{s_2}$ to estimate $\|s_1 - s_2\|^2$, our estimate of $r(S_1, S_2)$ is

$$\hat{r}(S_1, S_2) = \frac{|S_1| + |S_2| - \|\hat{s}_1 - \hat{s}_2\|^2}{|S_1| + |S_2| + \|\hat{s}_1 - \hat{s}_2\|^2}. \tag{12}$$

2.2.3. Mis-identification errors

Using $\hat{r}(S_i, S_j)$ instead of $r(S_i, S_j)$, introduces errors in identifying measurement co-occurrence with threshold τ. Such errors happen when $\hat{r}(S_i, S_j) < \tau$ while $r(S_i, S_j) \geqslant \tau$ (false positive errors) and when $\hat{r}(S_i, S_j) \geqslant \tau$ while $r(S_i, S_j) < \tau$ (false negative errors). The number of such errors depends on k, the number of hash functions used in the positional min-wise hashing or the number of dimensions used for the random projections, the threshold τ, and the distribution of the values $r(S_i, S_j)$ over all the sets S_i, S_j.

Theorem 2 provides an upper bound on the probability of such mis-identification errors when using random projections (similar estimates can be obtained using Theorem 1 and the proof of Theorem 2).

Theorem 2. *Consider a family of* m *sets* $S_1, S_2, \ldots, S_m \subseteq [0, w)$, *and a threshold* $0 \leqslant \tau \leqslant 1$ *for identifying co-occurrence among any pair of them. Let* $0 < \delta < 1$ *and* $\beta \geqslant 0$. *The probability of an error in identifying co-occurrence between any pair of sets* S_i, S_j *when using* $\hat{r}(S_i, S_j)$, *with random projections onto k vectors, instead of* $r(S_i, S_j)$ *is at most* $m^{-\beta}$ *plus the probability that* $r(S_i, S_j) \in [(1 - \delta)\tau, (1 + \delta)\tau]$, *where* $k \geqslant \frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3}\log m$, $\epsilon \leqslant \min\{\ \zeta^{-1} - 1,\ 1 - \zeta\ \}$, *and* $\zeta = \frac{1-\tau}{1+\tau} \cdot \frac{1+(1-\delta)\tau}{1-(1-\delta)\tau}$.

Proof. The proof of this theorem uses Lemmas 2 and 3 given in Appendix C. Using $a = |S_i| + |S_j|$ and $b = \|s_i - s_j\|^2$ in Lemma 3, we get a band (range) of resemblance values $I = [(1 - \delta)\tau, (1 + \delta)\tau]$ together with an upper bound $\epsilon_0 = \min\{\ \zeta^{-1} - 1,\ 1 - \zeta\}$ on ϵ, such that, $\hat{r}(S_i, S_j) \notin I$ if and only if $r(S_i, S_j) \notin I$, provided that $\|\hat{s}_i - \hat{s}_j\|^2$ is within $1 \pm \epsilon$ of $\|s_i - s_j\|^2$. The latter happens with probability $1 - m^{-\beta}$ for $k \geqslant \frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3}\log m$, as given by Lemma 2. Thus, with probability $1 - m^{\beta}$ there are no errors in identifying co-occurrence when the true resemblance is outside the band I. Therefore, the probability of an error in identifying co-occurrence when using $\hat{r}(S_i, S_j)$ instead of $r(S_i, S_j)$ is at most $m^{-\beta}$ plus the probability that $r(S_i, S_j) \in [(1 - \delta)\tau, (1 + \delta)\tau]$. □

For example, for $\tau = 0.95$, $\delta = 10^{-2}$, $\beta = 1$, and uniform distribution of true resemblance over $[0, 1]$, the probability of co-occurrence mis-identification errors for the random projections approach is $\leqslant 0.03$ provided that $k \geqslant 502\log m$. In our experimental results, we find that much smaller values of k are sufficient for small error $|r - \hat{r}|$ (note that here $m \leqslant 2w$). Furthermore, we experimentally find that the resemblance estimation error of two sets by using the random projection approach is $\approx 50\%$ smaller than that obtained with the positional min-wise hashing approach.

2.2.4. Element signatures

We define the *(positional) min-wise signature* of an element e within window W to be the (positional) min-wise hash of its occurrence set $\chi_W(e)$. Similarly, the *random projection signature* of e is the random projection of $\chi_W(e)$. 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, while the time to compute it is $O(kw)$.

3. Collaborative data gathering protocol exploiting measurements co-occurrence

We present a protocol that the sensors and the base station in a wireless sensor network 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 and to collaborate by 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, larger than the window size w.

Sensors identify pairs of co-occurring elements e, e' and notify the base station that e, e' are co-occurring. The base station maintains the co-occurrence (symmetric) relation $C : U \times U \rightarrow \{0, 1\}$, such that $C[e_1, e_2] = 1$ iff it has been notified that e_1 and e_2 co-occur, where $U = \cup_{i=1}^{n} U_i$ is the universe of all the sensor elements. The binary relation C at the base station is represented efficiently using standard data structures for sparse undirected graphs. Further, a sensor i may choose, at any time, a default element $m_{\text{default}}(i) \in U_i \cup \{\textbf{null}\}$ among its non-co-occurring elements, and notify the base station of its choice. A sensor i does not communicate $m_{\text{default}}(i)$ to the base station each time it measures $m_{\text{default}}(i)$. The base station maintains the set of $m_{\text{default}}(i)$ it has been notified of.

At the end of each time (discrete period) t, the base station makes an inference (estimate) $\hat{m}_{i,t}$ of the value $m_{i,t}$ sensor i measures at t. For each sensor i and time t, we define the candidate measurements $V_{i,t}$ to be a list of those elements $e = (i, v) \in U_i$ that co-occur with an element $e\prime \in U - U_i$ communicated to the base station at time t. By default, $V_{i,t}$ is considered as a FIFO list (i.e. the elements e are ordered according to the order of arrival of their co-occurring elements e' at the base station). We define the extended candidate measurements $\widehat{V}_{i,t}$ to be equal to $V_{i,t}$ if $V_{i,t} \neq \emptyset$, equal to $(m_{\text{default}}(i))$ if $V_{i,t} = \emptyset$ and $m_{\text{default}}(i) \neq \textbf{null}$, and otherwise to be equal to the list of elements $e \in U_i$ that co-occur with an element $e\prime \in U - U_i$ (in any order). Observe that $\emptyset \subset \widehat{V}_{i,t} \subseteq U_i$. The base station makes an *inference error* if $\hat{m}_{i,t} \neq m_{i,t}$.

The choice of the element in $\widehat{V}_{i,t}$ used to compute $\hat{m}_{i,t}$ affects the magnitude $|\hat{m}_{i,t} - m_{i,t}|$ as well as the likelihood of an inference error. For simplicity we choose the first element in $\widehat{V}_{i,t}$. Alternate choices of interest would be (a) the value of the element with highest estimated resemblance in $V_{i,t}$,[3] (b) the value of the most frequent element in $\widehat{V}_{i,t}$, (c) the median value of the elements in $\widehat{V}_{i,t}$, or (d) the frequency-weighted average of the values in $\widehat{V}_{i,t}$, i.e.

$$\text{arc} \min_{v \in U_i} \left(\sum_{e \in \widehat{V}_{i,t}} f_e |v - e.v|^2 \right), \tag{13}$$

393

[3] The base station estimates co-occurrence of elements using its inferred measurements.

where f_e is the frequency of element e among i's measurements. Choice (a) may be attractive when attempting to reduce the likelihood of making an inference error, while the other choices may be attractive when attempting to reduce the magnitude of the inference errors.

The choice of $m_{\text{default}}(i)$ also affects both the magnitude and likelihood of inference errors. The default element $m_{\text{default}}(i)$ chosen by each sensor i is one of its most frequently occurring elements that does not co-occur with elements of other sensors. Our choice of $m_{\text{default}}(i)$ attempts to be aggressive on reducing the number of measurements communicated to the base station. Other choices of interest would be the frequency-weighted average or median value of the elements that do not co-occur with elements from other sensors if the magnitude of inference errors is of primary interest. Furthermore, if sensor i has an element e with high frequency but short $\Phi[e]$.list, then it should choose e as its default $m_{\text{default}}(i)$.

Sensor i maintains for each co-occurring element e a data structure $\Phi[e]$ that consists of: a list $\Phi[e]$.list of elements that have been identified as co-occurring with e (by sensor i or any other sensor), sorted in increasing order of their sid's; a list $\Phi[e]$.children of elements that have been identified as co-occurring with e by sensor i itself; an attribute $\Phi[e]$.state indicating the status of $\Phi[e]$. Attribute $\Phi[e]$.state takes values (a) *normal* if $\Phi[e]$ is up-to-date, (b) *waiting* if sensor i initiated an update and is waiting for acknowledgement messages, (c) *init* if sensor i had been the initiator of an update in the current window, (d) *updating* if sensor i received an update message UpdateMsg in the current window. For an element e that is not co-occurring with any other element, we assume $\Phi[e] = \textbf{null}$, and the sensor does not store $\Phi[e]$.

Furthermore, each sensor i maintains a set $R_i \subseteq U_i$ of elements that should always be communicated to the base station. The base station may ask sensor i to append an element e to R_i, e.g. if inference errors for such elements are unacceptable to the application.

At each time t, sensor i communicates to the base station its measuring element $e = (i, m_{i,t})$, if $e \neq m_{\text{default}}(i)$ or $e \in R_i$ or e does not co-occur with any other elements. If e co-occurs with other elements, sensor i may not need to communicate e to the base station, since the responsibility to communicate the co-occurring elements to the base station is shared among all the sensors in $\Phi[e]$.list. The current window is partitioned into $|\Phi[e].\text{list}|$ sub-windows, called *duty-zones*, with each sensor taking charge of one duty-zone. Sensor i will communicate e to the base station iff e occurs during a duty-zone for which sensor i is on duty. See Fig. 2 for further details.

We choose to split each window into equi-length duty-zones and let each sensor in the $\Phi[e]$.list take charge of a single duty-zone. This makes the scheduling simple and distributive, and enables sensors to quickly join or leave $\Phi[e]$.list. Different ways to split a window into duty-zones are possible. For example, in the equi-depth approach, the window is split into duty-zones so that each one has

```
SensorMeasurementLoop()
// Sensor i takes measurements
foreach window W do
    χ_W(e) ⟵ ∅ for all e ∈ U_i
    foreach time t within W do
        take a measurement v and create element e = (i, v)
        append relative time t̃ to χ_W(e)
        if e ∈ R_i or IsOnDuty(i, e, t̃) then
            send MeasurementMsg(e) to base station
    // end of window
    if t̃ = w − 1 then
        TSS_i ⟵ ∅
        foreach e ∈ U_i that occured in the current window W do
            σ_e ⟵ signature of χ_W(e)
            append the tuple (e, σ_e) to TSS_i
        if Φ[e].state ≠ WAITING then
            Φ[e].state ⟵ NORMAL
```

```
IsOnDuty(sid i, element e, relative time t)
// Sensor i computes its duty status for element e with e.sid = i at relative time t
if Φ[e] = null or Φ[e].state is WAITING or INIT then
    return true
else
    if e.value = m_default(i) or Φ[e].state = UPDATING then
        return false
    else
        let j be the index of the element e in Φ[e].list
        split the window [0, w − 1] into |Φ[e].list| intervals I_0, I_1, I_2, ...
            if t is in interval I_j then
                return true
            else
                return false
```

Fig. 2. The algorithm used by sensors to schedule transmissions of measurements to the base station.

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 $\Phi[e]$.list. Such an approach may lead to longer network lifetimes, provided the overhead in computing equi-depth duty zones is small. Also, sensors with more residual energy in $\Phi[e]$.list may take charge of multiple duty-zones. We defer consideration and comparison of such alternatives to future work.

Sensor i may initiate the discovery of co-occurring elements at any time, using the connected-components or the clique approach. We present the discovery algorithm with the connected-components approach in Fig. 3. Sensor i maintains a set TSS_i with the signatures of its elements, with respect to the previous window. The set TSS_i contains only those elements in U_i that occurred in the previous window at sensor i. Sensor i updates the set TSS_i of element signatures at the end of the current window. Sensor i requests the set TSS_j of signatures of elements from sensors $j \in Adj_i$, and if it finds among them an element e' that co-occurs with e then it (i) adds e' to $\Phi[e]$.list and $\Phi[e]$.children, and (ii) updates the base station and all the sensors with an element in $\Phi[e]$.list. Whenever a sensor receives such an update, it further updates all its children not already updated. See Fig. 3 for further details. Note that only elements e, e' with $e.sid \neq e'.sid$ may co-occur for any threshold $\tau > 0$. Further, since in practice $\tau > 1/2$, an element of sensor i can co-occur with at most one element from another sensor j, hence the break statement at line 10 in DiscoverCoOccurrences routine.

```
DiscoverCoOccurrences(threshold τ)
// Sensor i discovers co-occuring elements at threshold τ
1   foreach sensor j ∈ Adj_i do
2       request TSS_j from sensor j
3       foreach tuple (e, σ_e) ∈ TSS_i do
4           foreach tuple (e′, σ_e′) ∈ TSS_j do
5               if r(σ_e, σ_e′) ≥ τ then
6                   if Φ[e] = null then
7                       append e to the Φ[e].list
8                   append e′ to Φ[e].list and to Φ[e].children
9                   mark Φ[e] as changed
10                  break
11  foreach (e, σ_e) ∈ TSS_i do
12      if Φ[e] is marked as changed then
13          send NewMsg(Φ[e].list) to the base station
14          Φ[e].state ⟵ WAITING
15          foreach e′ ∈ Φ[e].list do
16              send UpdateMsg(i, Φ[e].list) to sensor e′.sid
17  upon receiving AckMsg for every UpdateMsg sent do
18      Φ[e].state ⟵ INIT
```

```
ReceiveUpdateMsg(sid j, set S)
// Sensor i receives UpdateMsg from sensor j
find the element e ∈ U_i ∩ S
Φ[e].state ⟵ UPDATING
append to Φ[e].list all the elements in S
foreach element e′ ∈ Φ[e].children do
    if e′ ∉ S then
        send UpdateMsg(i, Φ[e].list) to sensor e′.sid
upon receiving AckMsg for every UpdateMsg sent do
    send AckMsg to sensor j
```

Fig. 3. The algorithm used by sensors to discover element co-occurences.

Sensor i may decide to remove any of its co-occurring elements e from the co-occurrence relation at any time. In such a case, sensor i removes e from $\Phi[e]$.list, chooses a sensor j in $\Phi[e]$.list to act as a removal coordinator for updating the remaining sensors with elements in $\Phi[e]$.list. The removal coordinator sensor j "adopts" e's children $\Phi[e]$.children, and it tells all sensors with an element in $\Phi[e]$.list to remove e from their co-occurrence lists. See Fig. 4 for further details. Note that when sensor i was asked by the base station to append e to R_i, the base station may or may not ask i to remove e from co-occurrence relationship. It is advantageous for the other sensors with elements in $\Phi[e]$.list if the base station chooses not to ask i to remove e from the co-occurrence relation.

The base station may compute co-occurrence of elements using the inferred measurements $\hat{m}_{i,t}$ in a window, and then use this information to provide hints to the sensors to initiate either the discovery or removal of co-occurring elements. Such discovery or removal may be targeted since the base station could hint sensors to verify co-occurrence of specific pairs of elements. This will be useful, for example, when the communication costs of the discovery or removal become prohibitive (e.g. sensors with large number of neighbors). Moreover, in the extreme, we can have the base station compute all the co-occurrence relationships, together with choosing a default value and transmission schedule for each sensor to minimize the likelihood and/or magnitude of inference errors.

3.1. Analysis of the costs of the protocol

We analyze the worst-case running-time, memory, and communication costs of our protocol for collaborative data gathering.

Consider the BaseStationEstimator routine. Since $\widehat{V}_{i,t} \subseteq U_i$, it follows that the worst-case running-time for each time period t is $\sum_{i=1}^{n} O(U_i) = O(U)$. Note that, for the case where we use just the first element in $V_{i,t}$ to estimate $\hat{m}_{i,t}$, the worst-case running-time is reduced to $O(n)$ by skipping the computation of the complete set $\widehat{V}_{i,t}$. The memory required at the base station to store the co-occurrence relation C is $O(n+m)$, where $m = O(U^2)$ is the number of pairs of co-occurring elements.

Let $M = \max_{e \in U}\{|\Phi[e].\text{list}|\}$, $d = \max_{i=1}^{n}\{|Adj_i|\}$, $\gamma = \max_{i=1}^{n}\{\min\{w, U_i\}\}$, and $u = \max_{i=1}^{n}\{U_i\}$.

Observe that, for each window, $|TSS_i| \leqslant \min\{w, |U_i|\} \leqslant \gamma$.

Consider the SensorMeasurementLoop routine executed at sensor i. For each window, its worst-case running-time is

$$\begin{aligned}\sum_{(e,\sigma_e) \in TSS_i} & w \cdot O(\min\{w, \Phi[e].\text{list}\}) + O(TSS_i \cdot k \cdot w) \\ &= O(\min\{w, U_i\} \cdot (w \cdot \min\{w, M\} + k \cdot w)) \\ &= O((w \cdot \min\{w, M\} + k \cdot w) \cdot u), \end{aligned} \tag{14}$$

while the worst-case total memory required to maintain $\Phi[e]$, for all $e \in U_i$, is $O(U_i M) = O(M \cdot u)$. The memory required for TSS_i is $O((k+w) \cdot TSS_i) = O((k+w) \cdot \min\{w, U_i\}) = O((k+w) \cdot u)$.

```
Remove(element e)
// Sensor i removes e from its co-occurrence relationships with other elements
    send RemoveMsg(e) to base station
    choose a sensor j as remove coordinator among the sensors with elements in Φ[e].list
    send RemoveCoordinateMsg(e, Φ[e].children) to sensor j
    set Φ[e] ⟵ null

ReceiveRemoveCoordinateMsg(element e, set S)
// Sensor i receives RemoveCoordinateMsg
    find element e′ ∈ U_i that co-occurs with e, e.g. e ∈ Φ[e′].list
    remove e from both Φ[e′].list and Φ[e′].children
    append to Φ[e′].children all the elements in S
    Φ[e′].state ⟵ WAITING
    foreach e″ ∈ Φ[e′].list do
        send RemoveMsg(i, e, Φ[e′].list) to sensor e″.sid
        upon receiving AckMsg for each RemoveMsg sent do
        Φ[e′].state ⟵ INIT

ReceiveRemoveMsg(sid j, element e, set S)
// Sensor i receives RemoveMsg from sensor j
    find the element e′ ∈ U_i ∩ S
    Φ[e′].state ⟵ UPDATING
    remove e from both Φ[e′].list and Φ[e′].children
    foreach e″ ∈ Φ[e′].children − S do
        send RemoveMsg(i, e, S) to sensor e″.sid
    upon receiving AckMsg for every RemoveMsg sent do
        send AckMsg to sensor j
```

Fig. 4. Algorithm used by sensors to remove an element from its co-occurence relationships.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

Consider now the DiscoverCoOccurrences routine at sensor i. Its worst-case running-time is

$$\begin{aligned}&\sum_{j \in Adj_i} O(k \cdot TSS_j \cdot TSS_i) + \sum_{(e,\sigma_e) \in TSS_i} O(\Phi[e].\text{list}) \\ &\quad = O(d \cdot k \cdot \min\{w, U_i\} \min\{w, U_j\} + \min\{w, U_i\} \cdot M) \\ &\quad = O(d \cdot k \cdot u^2 + M \cdot u), \end{aligned} \tag{15}$$

while sending a total of at most $\min\{w, U_i\}n = O(un)$ UpdateMsgs each of size $O(M)$.

Finally, consider the cost of the Remove procedure for an element e. This routine sends at most n messages altogether, each of size $O(U + M)$.

4. Experimental evaluation

We discuss the results of our experiments on evaluating the effectiveness of our in-network data co-occurrence detection methods, and the energy efficiency of data gathering protocol.

4.1. Data sets and performance metrics

We use synthetic datasets, as well as real sensor measurements downloaded from the "James Reserve Data Management System" [1]. For the real datasets, we download from [1] the temperature and humidity measurements taken by 12 sensors deployed in James Reserve, California during a two day span (August 9 and 10, 2005). In our simulation, we round the original measurements x to round(x).

We generate synthetic datasets using three parameters: the window size w, the element frequency f in the window, and the true resemblance r between pairs of occurrence sets. The occurrence set of an element in a window is generated using a uniform distribution (i.e. select $f \cdot w$ numbers in $[0, w)$ with uniform distribution).

For evaluating our co-occurrence detection methods, we use two primary metrics: (1) the average resemblance estimation error $\mid r(S_1, S_2) - \hat{r}(S_1, S_2) \mid$, and (2) the size k of the signatures σ_e used by the resemblance estimation methods. For evaluating our data gathering protocol, we use (1) the rate at which the base station makes inference errors, calculated as the ratio of the number of inference errors over the total number of measurements made by all the sensors, and (2) the relative reduction $\mid M_{\text{dg-r}} - M_{\text{dg}} \mid / \mid M_{\text{dg}} \mid$ of the communication costs for data gathering, where $M_{\text{dg-r}}$ and M_{dg} are the total number of measurements communicated to the base station by all the sensors, with and without our proposed collaborative data gathering method, respectively.[4]

The communication overhead of the proposed scheme is due to the discovery and removal of co-occurring elements in our protocol. These overheads are proportional to the size of TSS_i and the number of sensors in each sensor's neighborhood Adj_i, while the size of TSS_i is proportional to the size of the element signatures σ_e and the number of elements in TSS_i. The communication overhead is typically no more than the cost of communicating all the measurements to the base station during a single window. As long as co-occurrences persist for a few windows, the overhead is small compared to the savings in the measurement communication costs.

4.2. Experimental results – synthetic datasets

To evaluate and compare the performance of our two in-network data co-occurrence detection methods, we generated 100 pairs of occurrence sets, for window sizes w ranging from 256 to 2048. In this experiment, the true resemblance r between pairs of occurrence sets is ≈ 0.95, while the element frequency f is 30%. Fig. 5 shows the resemblance estimation performance of the positional min-wise and random projection methods for different window and signature sizes, while Fig. 6 shows their performance for signature size $k = \lg w$ (i.e. using k random vectors or min-wise hash functions). As expected, larger values of k result in better resemblance estimation. It can be seen that a signature size of $k = \lg w$ works well for both the positional min-wise and random projections methods. Furthermore, we see that the random projection method gives more accurate resemblance estimates compared to the positional min-wise method. Based on these results, in the remaining experiments, we use the random projection method with signature size $k = \lg w$ for resemblance estimation.

Next, we examine the behavior of the resemblance estimation for different sizes of occurrence sets (i.e. elements with different frequencies of occurrence), as well as occurrence sets with different true resemblances. The results of these experiments are given in Fig. 7. We see that the random projection signatures provide good resemblance estimation as the element frequencies f change, for occurrence sets with true resemblance ≈ 0.95. We also see that the estimated resemblance converges to the true resemblance as the true resemblance of occurrence sets increases, for element frequency f fixed at 30%. These results are useful for the following two important reasons. Since co-occurrence can happen for elements of different frequencies, it is critical for the resemblance estimation to be rather insensitive to element frequencies. Further, having the resemblance estimation become more accurate for larger values of true resemblance, allows us to use the resemblance threshold τ as a lever for controlling the number of estimation errors at the base station, and to better exploit the tradeoff between number of errors and communication costs.

Next, we consider the connected-components and the clique approaches for identifying set with $n \geqslant 2$ co-occurring elements. These two approaches affect both the base station inference error rate as well as the communication

[4] Alternate communication cost models are possible, e.g. distance of sensors from base station, etc. Though such models may give even better savings of the communication costs, the simple unit cost model is sufficient to demonstrate the benefits of our approach.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

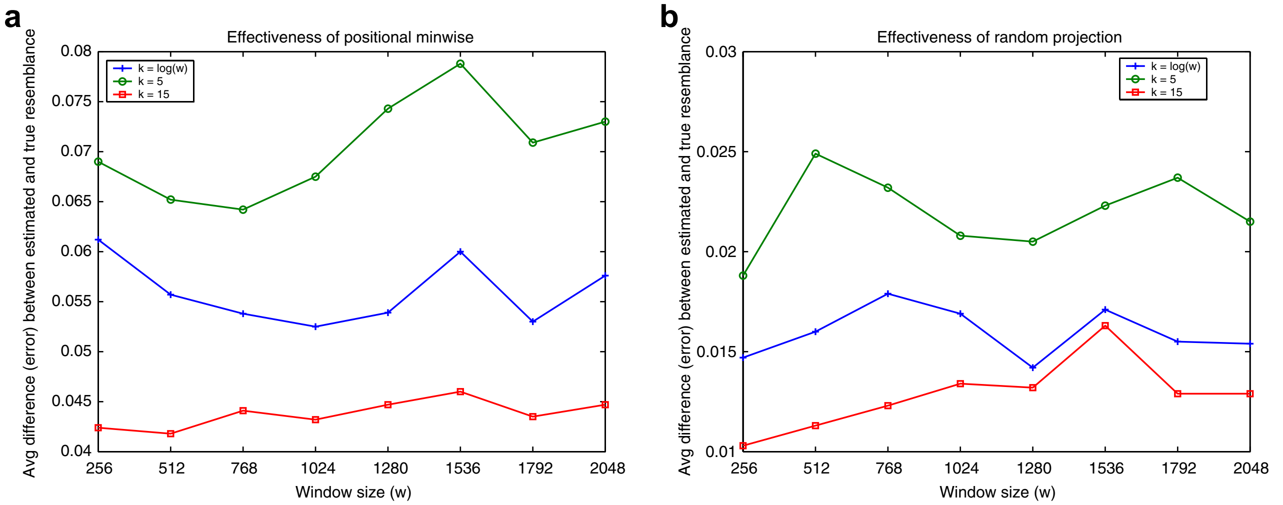

Fig. 5. Magnitude of the resemblance estimation error with varying signature size k, using (a) positional min-wise signatures, and (b) random projection signatures.

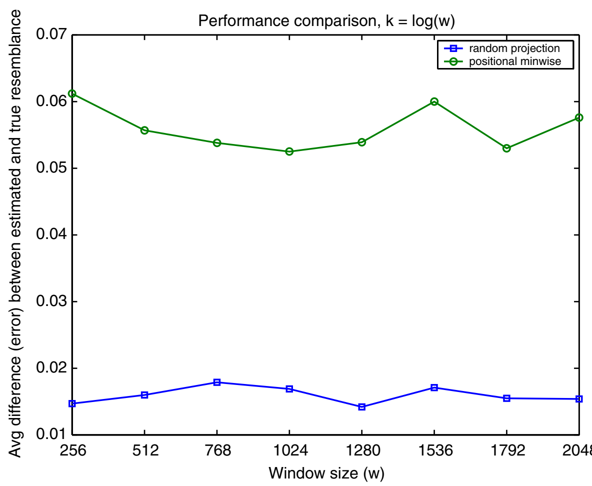

Fig. 6. Magnitude of the resemblance estimation error using positional min-wise and random projection signatures of size lg w.

costs of data gathering. For these experiments, we consider sets of $n = 2, 3, \ldots, 10$ elements, one element per sensor, with varying frequency f, window size $w = 1024$, true resemblance between pairs of element occurrence sets 0.95, signature size $k = \lg w$, and frequency of the default element chosen by each sensor equal to 30%. We apply both approaches for 100 consecutive windows. The results of these experiments are given in Figs. 8 and 9.

Fig. 8 gives the base station inference error rate, while Fig. 9 shows the relative reduction of communication costs. We can clearly see that the inference error rate is low, and that the reduction of communication costs is substantial and increases as n increases. Both achieve comparable communication cost savings when the same number of elements are identified as co-occurring. As expected, the inference error rate with the clique approach is lower and increases slower with n as compared with the connected-components

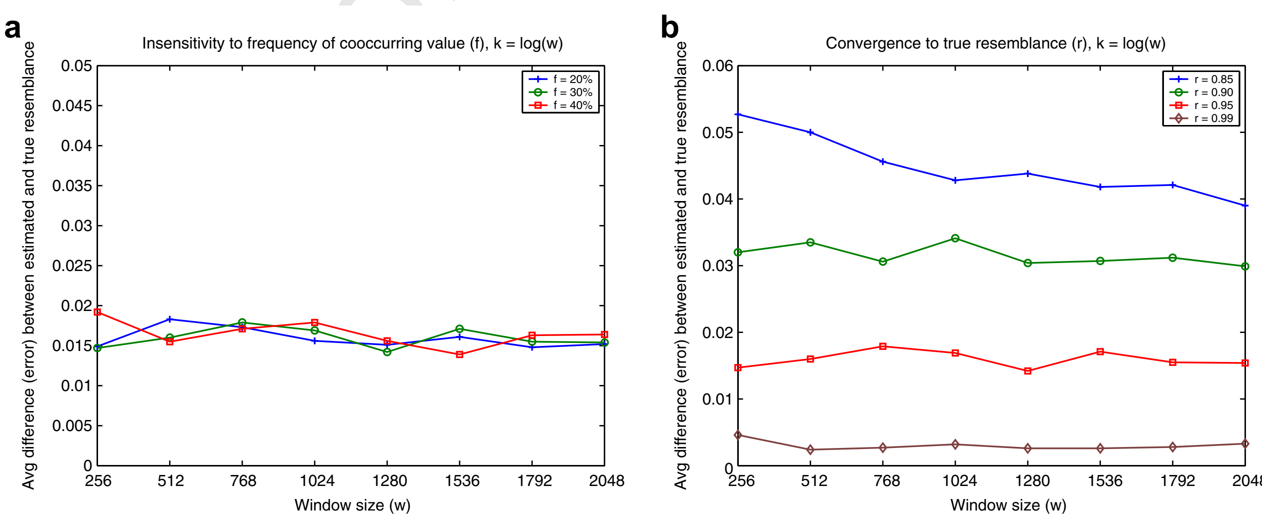

Fig. 7. Magnitude of the resemblance estimation error using random projection signatures of size lg w for (a) varying element frequencies and fixed true resemblance, and (b) varying true resemblance and fixed element frequency.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

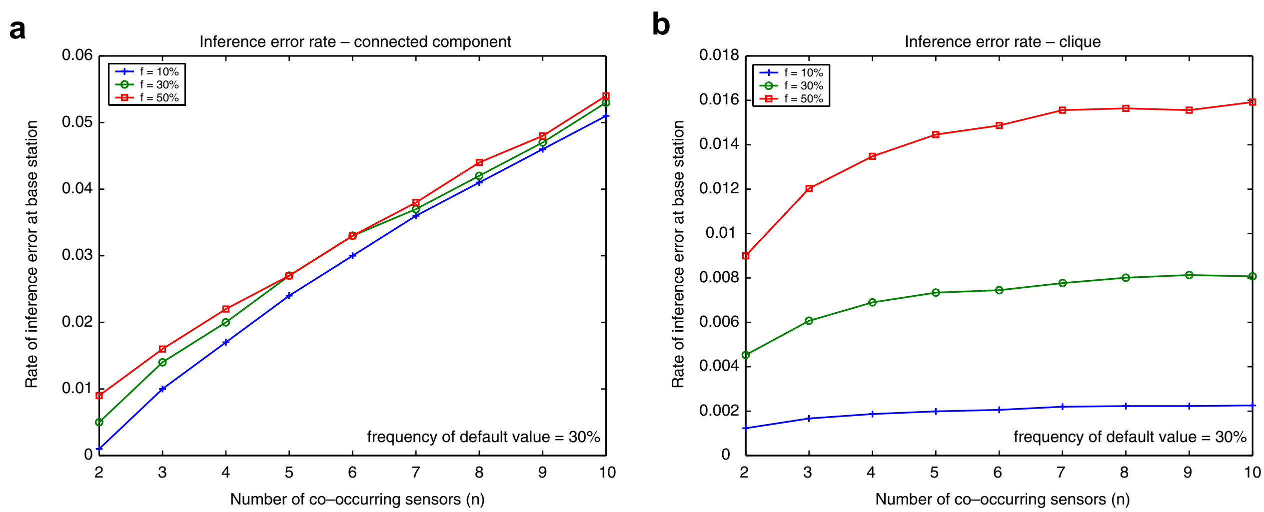

Fig. 8. Base station measurement inference error rate using (a) the connected-components approach and (b) the clique approach for identifying groups of co-occurring elements.

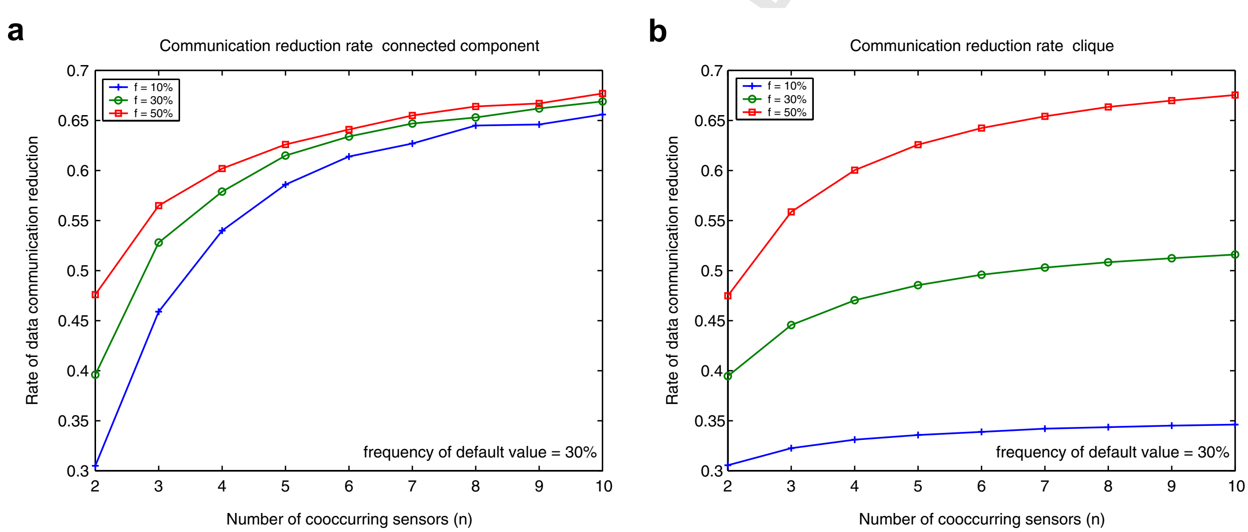

Fig. 9. Relative reduction of communication costs for collaborative data gathering when using (a) the connected-components approach, and (b) the clique approach for identifying sets of co-occuring elements.

approach. However, this advantage of the clique approach comes with higher communication and computation overheads, since the signatures of every element in $\Phi[e]$.list needs to be examined before an element e' can join $\Phi[e]$.list. Moreover, the clique approach may result in smaller $\Phi[e]$.list, leading into smaller reduction of communication costs. Consequently, the connected-components approach should be preferred for most applications.

4.3. Experimental results – real dataset

To evaluate how our method might perform in real sensor networks, we download the temperature and humidity measurements taken, every 5 min, by 12 sensors deployed in James Reserve, California during a two days span (August 9 and 10, 2005) and rounded each measurement x to $round(x)$.[5] We use window size $w = 512$, random projection signatures of size $\lg w$ with the connected-components approach for identifying sets of co-occurring elements, and resemblance threshold $\tau = 0.90$. We used the measurements in the first window for co-occurrence detection and found six sets of co-occurring elements among the temperature measurements: one with 6 elements, one with 3 elements, and four with 2 elements. Each sensor selects the most frequent temperature measurement (element) as its default element, among measurements (elements) that do not co-occur with measurements of other sensors. Following our method, during the first

[5] Data for only these two days were available. The dataset had some missing measurements. Each missing measurement was replaced with a unique new value.

window, the sensors communicated 4481 measurements out of 6144(12×512) measurements taken (a savings of 1663 measurements), while the number of measurement inference errors at the base station was 94. In other words, our scheme achieved a 27.1% reduction of the communication cost for data gathering with an error rate of 1.53%.

Though we are primarily interested in the number of inference errors rather than their magnitude, we note that for the real dataset, the p-norms of the all the sensor measurements $\|(\hat{m}_{i,t} - m_{i,t})_{i,t}\|_p$ and $\|(m_{i,t})_{i,t}\|_p$ are 58.20 and 1292.02 for $p = 2$, are 493 and 84562 for $p = 1$, and are 10 and 29 for $p = \infty$, respectively; the average and maximum of the relative error $\mid \hat{m}_{i,t} - m_{i,t} \mid / m_{i,t}$ over all i, t is 0.0068 and 0.7143. The magnitude of the inference errors made for the real data set is small.

5. Related work

Related work falls into two areas: set resemblance estimation and collaborative data gathering in wireless sensor networks.

5.1. Set resemblance estimation

Broder [4,5] utilizes min-wise independent hash functions for identifying near-duplicate documents on the web by estimating their resemblance using a fixed size signature for each document. Datar et al. [9] present min-wise based algorithms for estimating rarity and similarity on windowed data streams, accurate up to factor $1 \pm \varepsilon$ using space logarithmic in the window size. We also utilize a min-wise hashing as one of the methods for estimating resemblance of sets. However, to meet the computation constraints imposed by the sensors, we extend min-wise hashing to positional min-wise hashing to substantially reduce the required number of hash functions for computing signatures.

Random projections is a powerful dimensionality reduction technique with many applications, since it approximately preserves vector norms under some conditions. Cole et al. [7] utilize random projection on sketching the windowed time series data to discover Pearson correlation, for cases when orthogonal transformations such as DFT, DWT, or SVD can not be used because the data sets do not have any clear principal components. Similarly, Indyk et al. [15] use random projection on sketch computation for identifying representative trends in time series data. For stream data management, Thaper et al. [17] use random projection on constructing dynamic multidimensional histograms that succinctly approximate the data distribution of the underlying continue stream. We utilize random projection for estimating the resemblance of two remote sets.

Agarwal and Trachtenberg [3] propose protocols for estimating the number of differences between sets held on remote hosts, using counting Bloom filters [11]. Given the size of two sets from the same domain, the number of the differences between them can be calculated using their resemblance (see Section 2.2.2). Hence, our position min-wise hashing and random projection signatures are simple and effective methods for estimating the difference between remote sets from the same domain, at a cost which is logarithmic in the size of the domain.

5.2. Collaborative data gathering

Exploiting data correlations for data gathering in wireless sensor networks has been recently addressed by Cristescu et al. [8], Chou et al. [6], Rickenbach et al. [18], Sharaf et al. [16], Yoon and Shahabi [19], and Gupta et al. [13]. Cristescu et al. [8] consider the problem of finding the optimal transmission structure and the rate-distortion allocations at the various spatially distributed nodes, in order to minimize the total power consumption of the network. Chou et al. [6] and Rickenbach et al. [18] exploit correlations for coding the sensor measurements in order to reduce the total number of bits transmitted during data gathering. In Chou et al. [6], the data gathering node tracks the correlation structure among the sensor nodes, and uses this information to inform the sensors of the number of bits they should use for encoding their measurements. A fixed correlation structure is assumed, and all sensors are engaged in all of the data transmissions. In particular, Rickenbach et al. [18] consider foreign-coding and self-coding schemes and present algorithms for constructing optimal and near-optimal data gathering trees for foreign-coding and self-coding, respectively.

Gupta et al. [13] propose algorithms to select a subset of sensors, called connected correlation-dominating set, that form a connected communication graph and whose data may be sufficient to reconstruct data for the entire sensor network at the base station. During data gathering only those selected sensors will be involved in communication of measurements to the base station. They assume that sensors know the correlation structure, and their focus is on computing the correlation-dominating set. We present a scheme for sensors to detect, in-network, and then utilize measurement co-occurrence, which is different from correlation, for reducing the communication costs of data gathering.

Sharaf et al. [16] present the TiNA mechanism that reduces data transmissions and provides approximate results to aggregation queries by utilizing data temporal coherency. Sensors will send a reading upwards on an aggregation tree only if their reading differs from the last recorded reading by more than a given tolerance. Yoon and Shahabi [19] present CAG, a similar mechanism to TiNA, that utilizes spatial correlation of sensor data. When generating an aggregation tree, CAG forms clusters of nodes sensing values within a user-provided error tolerance. Subsequently only one value per cluster is transmitted upwards on the aggregation tree. Both TiNA and CAG exploit data correlation in the context of in-network aggregation, while our work utilizes data co-occurrence on gathering every measurement from all the sensors.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

Moreover, while these works above that utilize data correlation need numerical data, and it is unclear how they should handle enumerated (non-numerical) data, our approach works for both discrete numerical and non-numerical data.

Statistical models have been used for data acquisition in [12,10]. Goel et al. [12] present PERMON, a system for motion detection using the spatio-temporal correlation in sensor readings to reduce data transmissions. In PERMON, the base station generates a prediction model for each sensor based on sensor readings, and it sends these models back to the sensors. After receiving a prediction model, sensors will send a new reading only when it differs from the one in the motion prediction model. Deshpande et al. [10] incorporate parametric statistical models of the real-world into their sensornet query processing architecture. Time-varying multivariate Gaussian models are used, and sensors will be used to acquire and transmit data only when those models are insufficient to answer queries with acceptable accuracy. In a sense, 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 [10] does.

6. Conclusion

We describe a novel approach for the problem of gathering at a base station all the measurements of the sensors in wireless sensor networks. Our approach exploits co-occurrence of sensor measurements, in order for sensors to share the cost of communicating co-occurring measurements to the base station. The base station makes inferences, sometimes erroneous, about the true sensor measurements based on the information it receives. We present two energy efficient methods enabling the sensors to estimate the resemblance of their measurement occurrence sets, one using (positional) min-wise hashing and one using random projections. We also present a simple and effective protocol for sensors to collaborate in transmitting measurements to the base station.

We experimentally evaluate the proposed approach, and find that it offers substantial savings in the communication costs for few number of inference errors at the base station. For synthetic datasets it provides up to 65% savings on the communication costs and <6% inference errors, and for a real dataset it provides 27% savings and 1.53% inference errors.

Appendix A

Lemma 1. *For any two sets S_1, S_2 with resemblance $0 \leqslant p \leqslant 1$, $\min\{|S_1|, |S_2|\}/\max\{|S_1|, |S_2|\} \geqslant p/(1+p)$. Further, if $S_1 \neq \emptyset$ then $|S_1 \cap S_2| / |S_1| \geqslant p(1+2p)/(1+p)^2$.*

Proof. Assume, w.l.o.g., that S_1 is not empty. By definition,

$$p = r(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{16}$$

Since $|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$, we have that

$$\frac{|S_1 \cap S_2|}{|S_1| + |S_2|} = \frac{p}{1+p}. \tag{17}$$

Furthermore, since

$$\begin{aligned} \min\{|S_1|, |S_2|\} \geqslant |S_1 \cap S_2| &= \frac{p}{1+p}(|S_1| + |S_2|) \\ &\geqslant \frac{p}{1+p}\max\{|S_1|, |S_2|\}, \end{aligned} \tag{18}$$

it follows that

$$1 \geqslant \frac{\min\{|S_1|, |S_2|\}}{\max\{|S_1|, |S_2|\}} \geqslant \frac{p}{1+p}. \tag{19}$$

In addition, if $S_1 \neq \emptyset$ then

$$\begin{aligned} \frac{|S_1| + |S_2|}{|S_1|} &= 1 + \frac{|S_2|}{|S_1|} \geqslant 1 + \frac{\min\{|S_1|, |S_2|\}}{\max\{|S_1|, |S_2|\}} \\ &= 1 + \frac{p}{1+p} = \frac{1+2p}{1+p}, \end{aligned} \tag{20}$$

and the lemma follows. □

Appendix B

We construct sequences with co-occurring values of high frequency, and correlation coefficients that can be anywhere in the range $(-1, 1)$, thus demonstrating that the correlation coefficient is not an indicator of co-occurrence. Let $x(I)$ be the set of elements of a vector x with indices in an index set I, and let $x(I) = c$ indicate the fact that all elements $x(I)$ are equal to c. For a given frequency f, we generate a set I of $n \cdot f$ uniformly distributed random integers in $[1, n]$. Let $J = [1, n] - I$. We create the following six n-dimensional vectors:

- x_1 with $x_1(I) = 1$ and $x_1(J) = 0$.
- x_2 with $x_2(I) = 3$ and $x_2(J) = 0$.
- y_1 with $y_1(I) = 0$ and $y_1(J)$ be random real number uniformly distributed over $[0, 5]$.
- y_2 with $y_2(I) = 0$ and $y_2(J)$ be random real number uniformly distributed over $[-5, 0]$.
- $z_1 = x_1 + y_1$, and
- $z_2 = x_2 + y_2$.

Observe that 1 and 3 co-occur between x_1 and x_2, as do 1 and 0 between x_1 and y_1, etc. We then normalize each one of these six sequences to have mean 0 and variance 1. Every pair of the normalized sequences still has a pair of (discretized) values that co-occur (at a level ≈ 1.0 with high probability). The correlation coefficients between a sample of the sequences x_1, x_2, y_1, y_2, z_1, and z_2 for $n = 1000$ and $f = 0.95$ are

$$\begin{array}{cc} & \begin{array}{cccccc} \quad x_1 & \quad x_2 & \quad y_1 & \quad y_2 & \quad z_1 & \quad z_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z_1 \\ z_2 \end{array} & \begin{pmatrix} 1.0000 & 1.0000 & -0.8043 & 0.8034 & -0.6298 & 0.9459 \\ 1.0000 & 1.0000 & -0.8043 & 0.8034 & -0.6298 & 0.9459 \\ -0.8043 & -0.8043 & 1.0000 & -0.6265 & 0.9681 & -0.7500 \\ 0.8034 & 0.8034 & -0.6265 & 1.0000 & -0.4803 & 0.9532 \\ -0.6298 & -0.6298 & 0.9681 & -0.4803 & 1.0000 & -0.5817 \\ 0.9459 & 0.9459 & -0.7500 & 0.9532 & -0.5817 & 1.0000 \end{pmatrix} \end{array}. \tag{21}$$

Here each pair of these sequences has a pair of co-occurring values at the level of 1.0. The correlation coefficient between pairs of distinct sequences does not indicate whether any values co-occur.

Appendix C

Lemma 2 (Achlioptas [2]). *For every set* $S = \{s_1, s_2, \ldots, s_n\} \subseteq R^d$, *and every* $\epsilon > 0$, *the projection* $\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n\}$ *onto a set of k vectors* $\{z_1, z_2, \ldots, z_k\} \subseteq R^d$, *each of whose entries* $z_{i,j}$ *are i.i.d. random variables, is such that for all* $1 \leqslant i, j \leqslant n$,

$$(1-\epsilon)\|s_i - s_j\| \leqslant \|\hat{s}_i - \hat{s}_j\| \leqslant (1+\epsilon)\|s_i - s_j\|, \tag{22}$$

with probability at least $1 - n^{\beta}$, *provided that* $k \geqslant \frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3}\log n$. *Each* $z_{i,j}$ *is an i.i.d. random variable that takes the values* $\sqrt{3/k}$, $-\sqrt{3/k}$, *and* 0 *with probability* 1/6, 1/6, *and* 2/3, *respectively.*

Similar results to Lemma 2 are known when the entries of the projection vectors are i.i.d. normal random variables $N(0,1)$, scaled by $1/\sqrt{k}$, where $k = \Omega(\epsilon^{-2}\log n)$.

Lemma 3. *Consider the function* $f(x) = (a - bx)/(a + bx)$ *for* $x \geqslant 0$, *where* $a > 0$, $b \geqslant 0$. *Function f is non-increasing in x, and* $f(1+\epsilon) \leqslant f(x) \leqslant f(1-\epsilon)$ *for* $1-\epsilon \leqslant x \leqslant 1+\epsilon$, *where* $0 < \epsilon < 1$. *Moreover, for all* $0 < \delta < 1$ *and* $0 < \lambda_1 < 1$,

$$\begin{aligned} f(0) \geqslant \lambda_1 \rightarrow f(1+\epsilon) \geqslant (1-\delta)f(0) \text{ and } f(1-\epsilon) \\ \geqslant \lambda_1 \rightarrow f(0) \geqslant (1-\delta)\lambda_1, \end{aligned} \tag{23}$$

if $\epsilon \leqslant \epsilon_0 = \min\{\ \zeta^{-1} - 1,\ 1 - \zeta\ \}$ *where* $\zeta = \frac{1-\lambda_1}{1+\lambda_1} \cdot \frac{1+(1-\delta)\lambda_1}{1-(1-\delta)\lambda_1}$.

Proof. Observe that for any $0 < \lambda_1 \leqslant \lambda_2 < 1$

$$\lambda_1 \leqslant f(0) \leqslant \lambda_2 \leftrightarrow c_1 b \leqslant a \leqslant c_2 b, \tag{24}$$

where $c_i = (1+\lambda_i)/(1-\lambda_i)$, for $i = 1, 2$. First, we prove that $f(0) \geqslant \lambda_1$ implies $f(1+\epsilon) \geqslant (1-\delta)f(0)$ if

$$\epsilon \leqslant \frac{1+\lambda_1}{1-\lambda_1} \cdot \frac{1-(1-\delta)\lambda_1}{1+(1-\delta)\lambda_1} - 1. \tag{25}$$

Since $f(0) \geqslant \lambda_1$, we have that $a \geqslant c_1 b$, where $c_1 = (1+\lambda_1)/(1-\lambda_1)$. Since $a + (1+\epsilon)b \leqslant a + (1+\epsilon)\frac{a}{c_1}$ and $a - (1+\epsilon)b \geqslant a - (1+\epsilon)\frac{a}{c_1}$, we have that

$$\frac{a-(1+\epsilon)b}{a+(1+\epsilon)b} \geqslant \frac{c_1 - 1 - \epsilon}{c_1 + 1 + \epsilon}. \tag{26}$$

Therefore, it is sufficient to have

$$\frac{c_1 - 1 - \epsilon}{c_1 + 1 + \epsilon} \geqslant (1-\delta)\lambda_1 \leftrightarrow c_1 \frac{1-(1-\delta)\lambda_1}{1+(1-\delta)\lambda_1} \geqslant 1 + \epsilon, \tag{27}$$

which implies that it is sufficient to have

$$\frac{1+\lambda_1}{1-\lambda_1} \cdot \frac{1-(1-\delta)\lambda_1}{1+(1-\delta)\lambda_1} - 1 \geqslant \epsilon. \tag{28}$$

Second, we prove that $f(1-\epsilon) \geqslant \lambda_1$ implies $f(0) \geqslant (1-\delta)\lambda_1$ if

$$\epsilon \leqslant 1 - \frac{1-\lambda_1}{1+\lambda_1} \cdot \frac{1+(1-\delta)\lambda_1}{1-(1-\delta)\lambda_1}. \tag{29}$$

Since $f(1-\epsilon) \geqslant \lambda_1$, we have that $a \geqslant c_1(1-\epsilon)b = c_1' b$, where $c_1 = (1+\lambda_1)/(1-\lambda_1)$. Therefore, $a - b \geqslant a - a/c_1'$ and $a + b \leqslant a + a/c_1'$, which implies that

$$f(0) = \frac{a-b}{a+b} \geqslant \frac{a - a/c_1'}{a + a/c_1'} = \frac{c_1' - 1}{c_1' + 1}. \tag{30}$$

Thus, it is sufficient to require that

$$\frac{c_1(1-\epsilon) - 1}{c_1(1-\epsilon) + 1} \geqslant (1-\delta)\lambda_1, \tag{31}$$

which is equivalent to having

$$\begin{aligned} \frac{c_1(1-\epsilon)-1}{2} &\geqslant \frac{(1-\delta)\lambda_1}{1-(1-\delta)\lambda_1} \\ \leftrightarrow \epsilon &\leqslant 1 - \frac{1-\lambda_1}{1+\lambda_1} \cdot \frac{1+(1-\delta)\lambda_1}{1-(1-\delta)\lambda_1}. \end{aligned} \tag{32}$$

□

References

[1] James Reserve Data Management System. http://cens.jamesreserve.edu/jrcensweb/cmstest/CMS_env_data_list.php.
[2] D. Achlioptas, Database-friendly random projections: Johnson-lindenstrauss with binary coins, Journal of Computer and System Sciences 26 (2003) 671–687.
[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] A.Z. Broder, Identifying and filtering near-duplicate documents, CPM 2000, LNCS 1848, pp. 1–10, 2000.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001

[5] A.Z. Broder, M. Charikar, A.M. Frieze, M. Mitzenmacher, Min-wise independent permutations, Journal of Computer and System Sciences 60 (3) (2000) 630–659.

[6] 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.

[7] R. Cole, D. Shasha, X. Zhao, Fast window correlations over uncooperative time series, in: KDD '05: Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM Press, New York, NY, USA, 2005, pp. 743–749.

[8] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, On network correlated data gathering. in: Proceedings of the IEEE INFOCOM, 2004.

[9] M. Datar and S. Muthukrishnan, Estimating similarity and rarity over data stream windows, in: Proceedings of the 10th European Symposium on Algorithms, Rome, Italy, September 2002.

[10] 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.

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

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

[13] 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.

[14] P. Indyk, Stable distributions, pseudorandom generators, embeddings and data stream computation, in: FOCS, pp. 189–197, 2000.

[15] P. Indyk, N. Koudas, S. Muthukrishnan, Identifying representative trends in massive time series data sets using sketches, in: VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 363–372.

[16] 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.

[17] N. Thaper, S. Guha, P. Indyk, N. Koudas, Dynamic multidimensional histograms, in: SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data, ACM Press, New York, NY, USA, 2002, pp. 428–439.

[18] P. von Rickenbach and R. Wattenhofer. Gathering correlated data in sensor networks. in: DIALM-POMC'04: Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing, ACM Press, New York, NY, USA, 2004, pp. 60–66.

[19] 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.

Please cite this article in press as: K. Kalpakis, S. Tang, Collaborative data gathering in wireless sensor networks ..., Comput. Commun. (2008), doi:10.1016/j.comcom.2008.01.001