

Principal Component Aggregation for
Energy-efficient Information Extraction in
Wireless Sensor Networks

Yann-Aël Le Borgne, Jean-Michel Dricot, and Gianluca Bontempi

ULB Machine Learning Group
Department of Computer Science
Université Libre de Bruxelles
1050 Brussels – Belgium
Tel: +32 - 2 650 59 14

E-mail: {yleborgn, jdricot, gbonte}@ulb.ac.be

1 Introduction

An efficient in-network data processing is a key factor to enable wireless sensor networks (WSN) to extract insightful or critical information. Therefore, an important amount of research has been devoted over the last years to the development of data processing techniques suitable for sensor networks [16, 40]. WSN are known to be constrained by limited resources, in terms of energy, network data throughput, and computational power. The communication module is a particularly constrained resource since the amount of data that can be routed out of the network is inherently limited by the network capacity [32]. Also, wireless communication is an energy consuming task and it is identified in many situations as the primary factor of lifetime reduction [1]. The design of data gathering schemes that limit the amount of transmitted data is therefore recognized as a central issue for wireless sensor networks [16, 28, 32].

An attractive framework for the processing of data within a WSN is provided by data aggregation services, such as those developed at UC Berkeley (TinyDB and TAG projects) [23, 24], Cornell University (Cougar) [39], or EPFL (Dozer) [4]. These services aim at aggregating data within the network in a time- and energy-efficient manner and are suitable for networks connected to a base station, from which queries on sensor measurements are issued. In TAG or TinyDB, for instance, SQL-like queries interrogate the network to receive raw data or aggregates at regular time intervals. The underly-

ing architecture is a synchronized routing tree, along which data is processed and aggregated from the leaves to the root (i.e., the base station) [23, 24]. The interest of the approach is related to the ability of computing, *within the network*, some common operators like *average*, *min*, *max*, or *count*, thereby greatly decreasing the amount of data that needs to be transmitted over the network.

In this chapter, we show that the aggregation service principle can be used to implement a distributed data compression scheme based on Principal Component Analysis (PCA) [15]. PCA is a classic, multivariate data analysis technique which allows to represent data samples in a basis called the *principal component basis* (PC basis), where data samples are uncorrelated. When sensor measurements are correlated, which is often the case in sensor networks, PC basis allows to represent the sensor measurements variations with a reduced set of coordinates. This feature inspired recent work in the domain of data processing for sensor networks where PCA is used for tasks like approximate monitoring [22], feature prediction [3, 10], and event detection [12, 20]. However, it is worthy noting that what is common to all these approaches is that the transformation of the sensed data in the PC basis takes place in a centralized manner in the base station.

What we propose here is a *principal component aggregation* (PCA_g) scheme where the coordinates of the measurements in the PC basis are computed in a distributed fashion by means of the aggregation service. This approach extends previous work on data aggregation operators and presents

the following advantages. First, PCA provides varying levels of compression accuracies, ranging from constant approximations to full recovery of original data. It can be therefore be used to trade application accuracy for network load, thus making the principal component aggregation scheme scalable with the network size. Second, the PCAg scheme demands all sensors to send exactly the same number of packets during each transmission, thereby balancing the network load among sensors. Given that network load is strongly related to the energy consumption [30], we will show that the balanced loading increases the network lifetime as well.

The PCAg procedure is implemented as a three-stage process. First, a set of N measurements is collected at the sink from the whole set of sensors. Second, a set of q principal components are computed at the sink and distributed in the network. The third step is the sensing itself where each node computes the principal component scores in a distributed fashion along the routing tree. Experimental results based on a real world temperature measurement campaign illustrate that the PCAg allows a recovery of 90% of the data variance at the base station, while reducing the network load of up to 20%.

The remaining of this chapter is organized as follows. Section 2 introduces the notation and describes the principle of a WSN aggregation service. Section 3 presents the PCA and details its implementation in an aggregation service. Section 4 analyzes the tradeoffs between network load, network lifetime, and accuracy of approximations. A set of experimental results based

on a real world data set is reported and discussed in Section 5. Related work and possible extensions are presented in Section 6 while Section 7 concludes the chapter.

2 Data Aggregation in Sensor Networks

2.1 Network Architecture

Let us consider a sensor network architecture of p nodes whose task is to collect sensor measurements at regular intervals. Data is forwarded to a destination node referred to as *sink* or *base station*, assumed to benefit from higher resources (e.g., a desktop PC). Let $t \in \mathbb{N}$ denote the discretized time variable and $x_i[t]$ be the measurement collected by the sensor i , $1 \leq i \leq p$, at time t . At each time t , the p resulting measurements form a vector $\mathbf{x}[t] \in \mathbb{R}^p$. The sampling period is referred to as *an epoch*.

Since the communication range of the nodes is limited, the sink will generally not be in range of all the sensors. Therefore, the information has to be relayed from sources to the sink by means of intermediate nodes. Fig. 1 presents an example of a routing tree that collects the data from a set of sensors and forwards them to a *sink*.

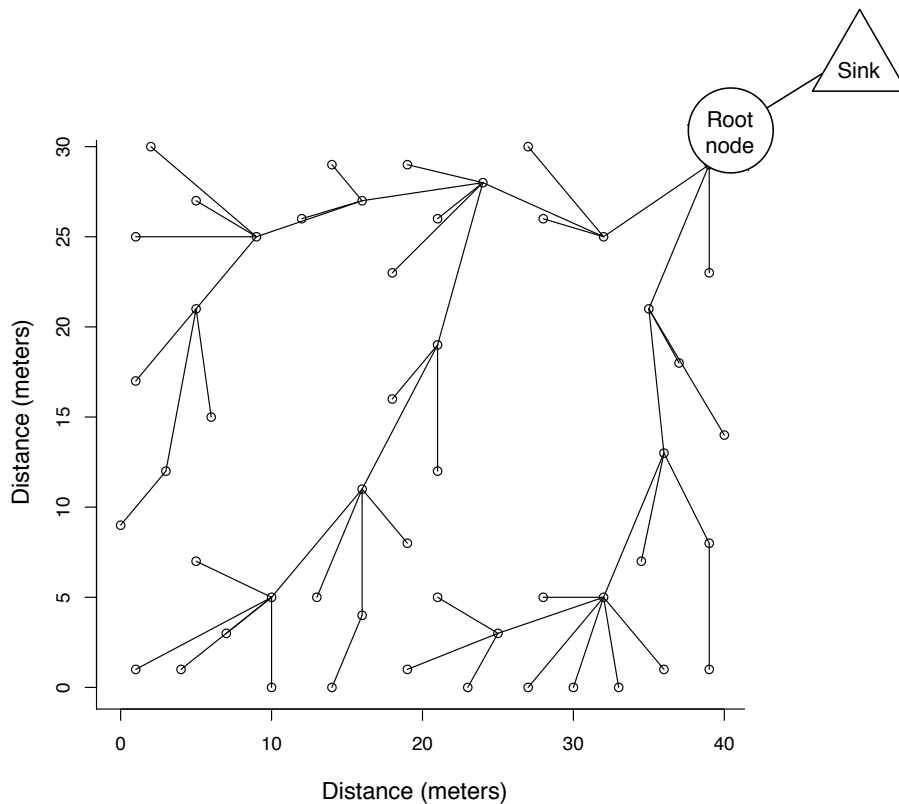


Fig. 1: Illustration of a routing tree connecting sensor nodes to a sink. Radio range is 10 meters.

2.2 Data Aggregation Service

This section presents an overview of *TAG*, a data aggregation service developed at the University of California, Berkeley [23, 24]. *TAG* stands for Tiny *AG*gregation and is an aggregation service for sensor networks which has been implemented in TinyOS, an operating system with a low memory footprint specifically designed for wireless sensors [33]. *TAG* aims at aggregating the data within the network in a time- and energy-efficient manner. To that

end, an epoch is divided into time slots, in such a way that the activities of the sensors are synchronized as a function of their depth in the routing tree. Any algorithm can be used to design the routing tree, as long as (i) it allows the data to flow in both directions of the tree, and (ii) it avoids sending duplicates [23].

The goal of TAG is to minimize the amount of time spent by sensors in powering their different components and to maximize the time spent in the idle mode, in which all electrical components are switched off. Indeed, the energy consumption is several orders of magnitude lower in the idle mode than in a mode where the CPU or the radio is active. This synchronization allows to significantly extend the lifetime of the sensors. An illustration of the activities of the sensors during an epoch is given in Fig. 2, for a network of four nodes with a routing tree of depth three.

Once a routing tree is set up and the nodes synchronized, data can be aggregated along the routing tree, from the leaves to the root. TAG relies on a set of three primitives [23, 24]:

- an initializer *init* which preprocesses a value measured by a sensor,
- an aggregation operator *f* which inserts the contribution of a node in the data flow, and
- an evaluator *e* which applies a final transformation on the data.

Each node includes its contribution in a *partial state record* X which is propagated along the routing tree. Partial state records are merged when two

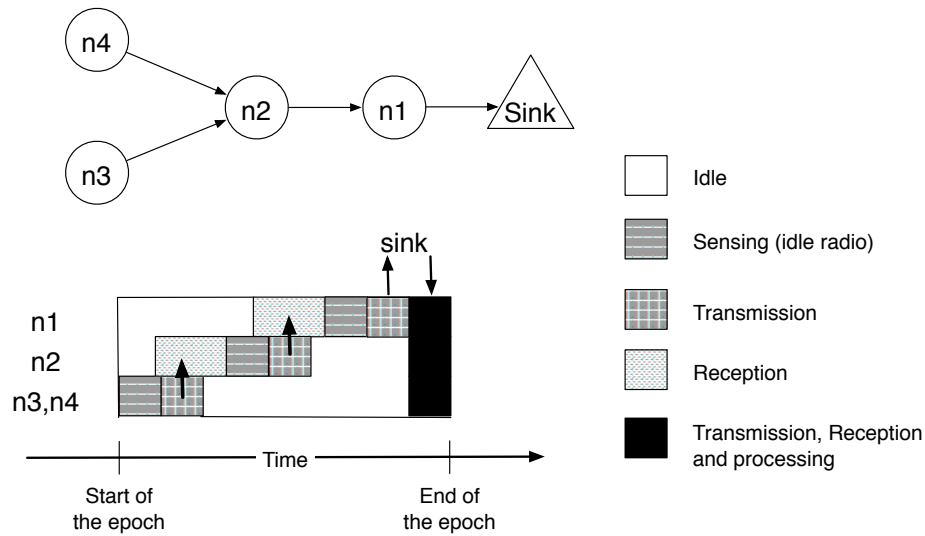


Fig. 2: Activities carried out by sensors depending of their depth in the routing tree (*adapted from [23]*).

(or more) of them arrive at the same node. When the eventual partial state record is delivered by the root node to the base station, the desired result is obtained thanks to the evaluator. Partial state records may be any data structures. However, when partial state records are scalars or vectors, the three operators defined above may be seen as functions.

Example: The 'average' aggregate can be computed with a partial state record $\langle x \rangle = (\text{SUM}, \text{COUNT})$ consisting of the sum of sensor measurements collected by nodes traversed, together with the number of nodes that contributed

to the sum. The three generic functions would be implemented as follows:

$$\begin{aligned} \mathit{init}(x_i[t]) &= \langle x_i[t], 1 \rangle \\ f(\langle S1, C1 \rangle, \langle S2, C2 \rangle) &= \langle S1 + S2, C1 + C2 \rangle \\ e(\langle S, C \rangle) &= S/C \end{aligned}$$

Note that without this aggregation process, all the measurements would be routed to the base station. The root node would therefore have to send p packets per epoch. Instead, using this scheme, each node is required to send only two pieces of data.

3 Principal Component Aggregation

3.1 Principal Component Analysis

The Principal Component Analysis (PCA) is a classic technique in statistical data analysis, data compression, and image processing [18, 25]. Given $q \leq p$ and a set of N centered¹ multivariate measurements $\mathbf{x}[t] \in \mathbb{R}^p$, it aims at finding a basis of q orthonormal vectors $\{\mathbf{w}_k\}_{1 \leq k \leq q}$ of \mathbb{R}^p , such that the mean squared distances between $\mathbf{x}[t]$ and their projections $\hat{\mathbf{x}}[\mathbf{t}] = \sum_{k=1}^q \mathbf{w}_k \mathbf{w}_k^T \mathbf{x}[t]$ on the subspace spanned by the basis $\{\mathbf{w}_k\}_{1 \leq k \leq q}$ is minimized. The corre-

¹measurements are centered so that the origin of the coordinate system coincides with the centroid of the set of measurements. This translation is desirable to avoid a biased estimation of the basis $\{\mathbf{w}_k\}_{1 \leq k \leq q}$ of \mathbb{R}^p towards the centroid of the set of measurements [18].

sponding optimization function can be expressed as:

$$\begin{aligned}
 J_q(\mathbf{x}[t], \mathbf{w}_k) &= \frac{1}{N} \sum_{t=1}^N \|\mathbf{x}[t] - \hat{\mathbf{x}}[t]\|^2 \\
 &= \frac{1}{N} \sum_{t=1}^N \left\| \mathbf{x}[t] - \sum_{k=1}^q \mathbf{w}_k \mathbf{w}_k^T \mathbf{x}[t] \right\|^2
 \end{aligned} \tag{1}$$

Under the constraint of orthonormal $\{\mathbf{w}_k\}_{1 \leq k \leq q}$, this expression can be minimized using the Lagrange multiplier technique [15]. The minimizer of (1) is the set of the q first eigenvectors $\{\mathbf{w}_k\}$ of the covariance matrix, ordered for convenience by decreasing eigenvalues λ_k . These eigenvectors are called the *principal components* and form the *principal component basis*. Eigenvalues quantify the amount of variance conserved by the eigenvectors, and their sum equals the total variance of the original set of centered observations X , i.e.:

$$\sum_{k=1}^p \lambda_k = \frac{1}{N} \sum_{t=1}^N \|\mathbf{x}[t]\|^2$$

The proportion P of retained variance within the first q principal components can be expressed as:

$$P(q) = \frac{\sum_{k=1}^q \lambda_k}{\sum_{k=1}^p \lambda_k} \tag{2}$$

Ranging columnwise the set of vectors $\{\mathbf{w}_k\}_{1 \leq k \leq q}$ in a $W_{p \times q}$ matrix, the

approximations $\hat{\mathbf{x}}[\mathbf{t}]$ of $\mathbf{x}[t]$ in the subspace \mathbb{R}^p are given by:

$$\hat{\mathbf{x}}[\mathbf{t}] = WW^T \mathbf{x}[t] = W\mathbf{z}[\mathbf{t}] \quad (3)$$

where

$$\mathbf{z}[\mathbf{t}] = W^T \mathbf{x}[t] = \begin{pmatrix} \sum_{i=1}^p w_{i1}x_i \\ \dots \\ \sum_{i=1}^p w_{iq}x_i \end{pmatrix} = \sum_{i=1}^p \begin{pmatrix} w_{i1}x_i \\ \dots \\ w_{iq}x_i \end{pmatrix}$$

denotes the column vector of the coordinates of $\hat{\mathbf{x}}[\mathbf{t}]$ in $\{\mathbf{w}_k\}_{1 \leq k \leq q}$, also referred to as the q *principal component scores*.

Example: Fig. 3 plots a set of $N = 50$ observations in a three dimensional data space x_1, x_2, x_3 where x_1, x_2 , and x_3 denote three data sources. Note that the correlation between x_1 and x_2 is high, while the x_3 signal is independent of x_1 and x_2 . The set of principal component (PC) basis vectors $\{w_1, w_2, w_3\}$, the two-dimensional subspace spanned by $\{w_1, w_2\}$ and the projections (crosses) of the original measurements on this subspace are illustrated in the figure. We can observe that the original set of three-variate measurements can be well approximated by the two-variate projections in the PC space, because of the strong correlations between the values $x_1[t]$ and the values $x_2[t]$.

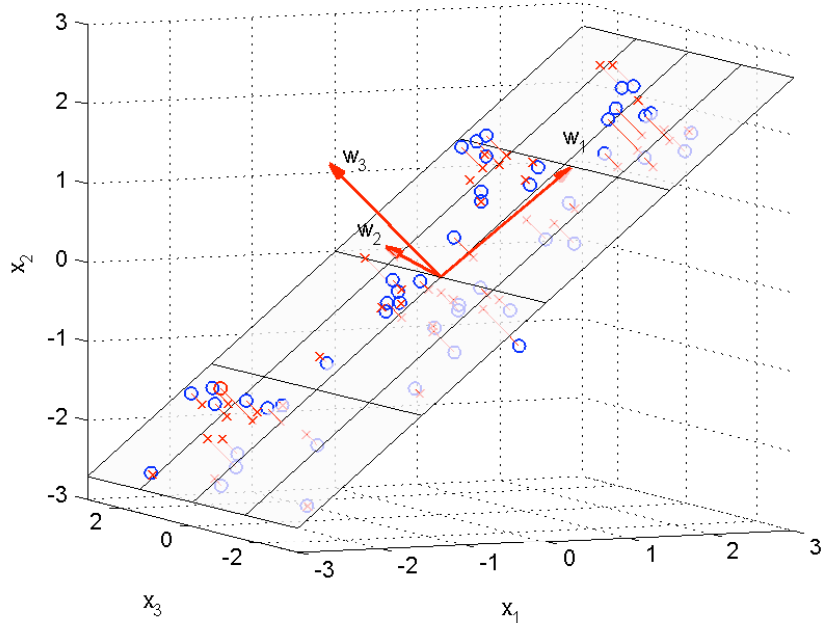


Fig. 3: Illustration of the transformation obtained by the principal component analysis. Circles denote the original observations while crosses denote their approximations obtained by projecting the original data on the two-dimensional subspace $\{w_1, w_2\}$ spanned by the two first principal components.

3.2 Implementation in a Data Aggregation Service

The computation of the q principal component scores $\mathbf{z}[\mathbf{t}]$ can be performed by an aggregation service if each node i is aware of the elements $\mathbf{w}_{i1}, \dots, \mathbf{w}_{iq}$ of the principal component basis. These elements are made available to each sensor during an initialization stage. The initialization consists in gathering at the sink a set of measurements from which an estimate of the covariance matrix is computed. The first q principal components are then derived and delivered to the network, so that each node i stores the elements $\mathbf{w}_{i1}, \dots, \mathbf{w}_{iq}$.

Note that the capacity of the principal components to properly span the signal subspace is dependent on the stationarity of the signal, and on the quality of the covariance matrix estimate. Failure to meet these two criteria may lead to poor approximations.

Once the components are made available to the network, the principal component scores are computed by the aggregation service, by summing along the routing tree the vectors $(\mathbf{w}_{i1}x_i[t], \dots, \mathbf{w}_{iq}x_i[t])$ available at each node. The aggregation primitives are:

$$\begin{aligned} \mathit{init}(x_i[t]) &= \langle \mathbf{w}_{i1}x_i[t]; \dots; \mathbf{w}_{iq}x_i[t] \rangle \\ f(\langle x_1; \dots; x_q \rangle, \langle y_1; \dots; y_q \rangle) &= \langle x_1 + y_1; \dots; x_q + y_q \rangle \end{aligned}$$

Partial state records are vectors of size q . The main characteristic of this approach is that each nodes *sends exactly the same amount of data*, i.e., the set of q coordinates $z_k[t]$.

3.3 Remote Approximation of the Measurements

An approximation \hat{x}_t of the measurements over the whole sensor field can be obtained at the base station by transforming the vector of coordinates z_t back to the original basis by using (3). The evaluator function is then the

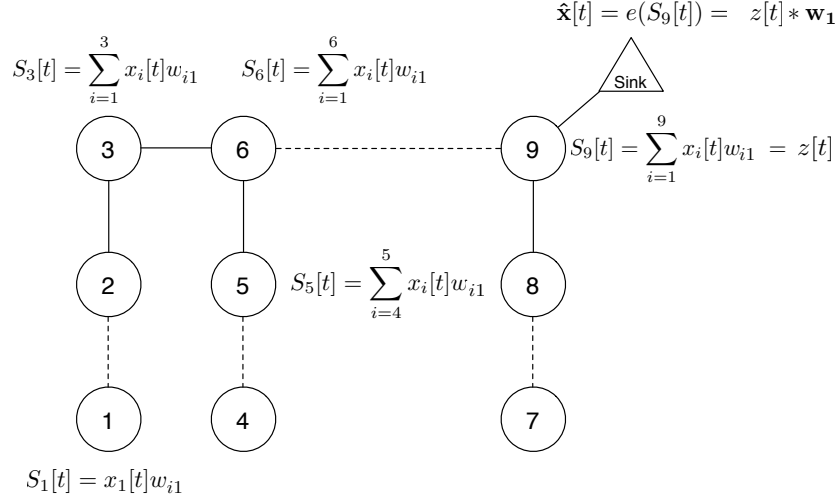


Fig. 4: Aggregation service at work for computing the projections of the set of measurements on the first principal component.

function

$$\begin{aligned}
 e(z_1[t], \dots, z_q[t]) &= (\hat{x}_1[t], \dots, \hat{x}_p[t]) \\
 &= W^T z[t]
 \end{aligned}$$

which returns the approximation of the p -variate sensor measurements by using the q principal components. Note that if $p = q$, the evaluation steps returns the exact set of sensor measurements. Otherwise, if the number of coordinates q is less than p , the evaluation will return an optimal approximations to the real measurements in the mean square sense (1). Since sensor measurements are often correlated, it is therefore likely that a number $q \ll p$ of coordinates can provide good approximations.

It is worthy noting that a simple procedure can be set up to check the accuracy of approximations with respect to a user defined threshold. According to (3) the approximation $\hat{x}_i[t]$ of the i th ($1 \leq i \leq p$) sensor measure at time t is given by:

$$\hat{x}_i[t] = \sum_{k=1}^q z_k[t] * w_{ik}$$

Since the terms $\{w_{ik}\}$ are assumed to be available at each node, each sensor is able to compute locally the approximation retrieved at the sink, and in case to send a notification when the approximation error is greater some user defined ϵ . This scheme, dubbed *supervised compression* in [21], guarantees that all data eventually obtained at the sink are within $\pm\epsilon$ of their actual measurements, and provides a way to decide when to update the principal components in case of non stationary signals.

4 Network Load and Energy Efficiency

This section presents an analysis of the impact of the principal component aggregation on the overall network performances. More precisely, we focus on the network traffic load, the distribution of the energy depletion among the nodes, and the scalability of the proposed solution. The scalability is defined as the capacity of the considered networking architecture to expand and adapt to an increasing number of sensor nodes [11]. This notion is of importance when considering large-scale deployments or very dense sensing scenarios. Also, in most networking systems, it is found to be a limiting

issue [19] and has therefore to be carefully evaluated.

We first address in Section 4.1 the tradeoff between the accuracy of the PCAg scheme and the gain in terms of network load. Next, we analyze in Section 4.2 the distribution of the network load in the case of the classical approach (i.e., store-and-forward) and with the PCAg. Finally, in Section 4.3 we conduct a detailed computation of the energy consumption in a scenario where a hierarchical routing topology [14,35] is used. A quantification of the expected gains, in terms of network load and scalability, is also presented in this section.

4.1 Tradeoff between Accuracy and Network Load

As discussed in Section 3.3 the data reconstruction carried out at the network sink provides an approximation of the sensed measurements. The precision of this approximation depends on the number q of principal components retained. At the same time, since q is also the number of components which needs to be transmitted over the wireless network by the aggregation service, the value of q has a direct impact on the network load.

In quantitative terms, Equation (2) illustrates the relation between the percentage of retained variance and the number of principal components:

$$P(q) = \frac{\sum_{k=1}^q \lambda_k}{\sum_{k=1}^p \lambda_k}$$

As eigenvalues are necessarily positive, the function $P(q)$ varies monotonically

cally with the value of q . Therefore, any *decrease* of the number of principal components results into a lower network load at the cost of an accuracy loss. On the other hand, an *increase* of the number of principal components has a positive effect on the amount of retained variance (and consequently on the sensing accuracy) but demands additional data to be transmitted. Therefore, the PCA scheme incurs a *tradeoff* between the reduction of the network load and the sensing accuracy.

Before detailing further how to formulate this tradeoff, we recall that the amount of information retained by a set of principal components depends on the degree of correlation among the data sources. Whenever nearby sensors collect correlated measurements, a small set of principal components is likely to support most of the variations observed by the network. As an example, we refer the reader to the Fig. 10 in the experimental section, which illustrates the relation between the percentage of variance retained and the number of principal components.

In practical settings, the benefits obtained in accuracy by adding a component must be weighted by the cost incurred in terms of network load. The weighting is necessarily application dependent, and can be formulated by means of an optimization function. Its optimum may be determined for example at the sink, by means of a cross validation procedure on the measurements collected during the initialization stage.

Finally, we emphasize that the principal component aggregation scheme is not appropriate when sensor measurements are not correlated, or if the

number of components required by the application is too high. We detail this aspect in the next section, and derive an upper bound on the number of principal component above which the default scheme should be preferred.

4.2 Distribution of the Network Load

Let us consider a generic routing tree, where each node of the topology relays the information from its children.

We begin by analyzing a classical store-and-forward (S/F) routing protocol [36] where each node receives Rx ($0 \leq Rx \leq p - 1$) packets from its children and p is the total number of nodes in the network. In particular, if the node is a leaf it does not receive any packet to forward ($Rx = 0$) while if the node is fully connected it receives $Rx = p - 1$ measurements per epoch.

After the reception of Rx measurements, a node adds its own data and forwards the whole set to its parent node. It will therefore forward $Tx = Rx + 1$ packets, where $1 \leq Tx \leq p$. It follows that the upper bound on the network load for all nodes of the topology is given by:

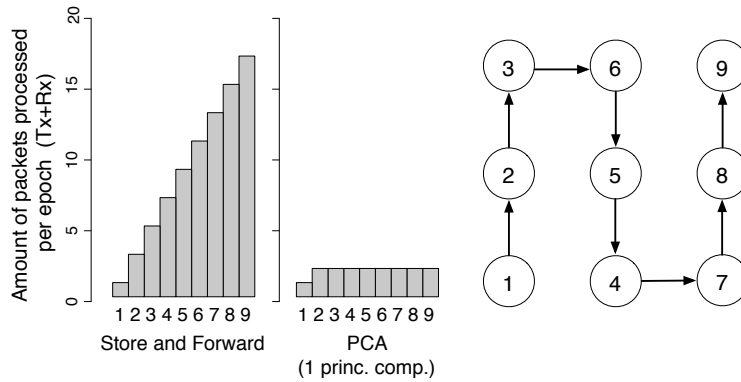
$$\begin{aligned} L &= \max_i \{Rx_i + Tx_i\} \\ &= (p - 1) + p = 2p - 1 \end{aligned} \tag{4}$$

where the subscript i refers to the i -th node in the network. The upper bound for the network load is a network metric that characterizes the minimum throughput required at network nodes for avoiding congestion issues [5].

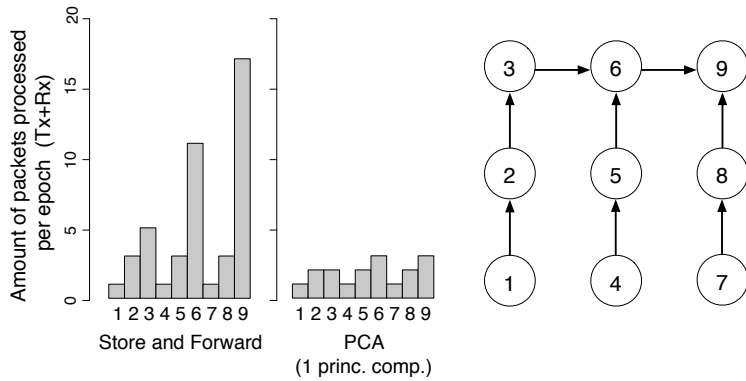
Let us consider now what happens when the PCAg is adopted. Each node receives q components from each of its neighbours. The total number of packets received by all nodes in the network is therefore $qC_{\min} \leq \text{Rx} \leq qC_{\max}$, where C_{\min} and C_{\max} stand for the minimum and the maximum number of children of nodes in the network. Since the data received by a node is combined with its sensed observation into a q -sized vector, the total number of packets forwarded by a node is equal to q . It follows that the upper bound on the network load of a node by using the PCA is:

$$\begin{aligned} L^{(\text{pca})} &= \max_i \{\text{Rx}_i + \text{Tx}_i\} \\ &= qC_{\max} + q = q(C_{\max} + 1) \end{aligned} \tag{5}$$

Fig. 5 reports bar plots of the per-node network load sustained for two different routing trees, and compares the network load distribution entailed by the S/F and the PCAg approaches. More precisely, Fig. 5(a) illustrates the repartition of the network loads in the case of a linear chain, while Fig. 5(b) refers to a more generic, hierarchical network tree. We remark that in the S/F approach the network loads sustained by the nodes are very heterogeneous. In fact, the load depends on the node position in the routing tree: a leaf node transmits only its own sensing information while the other nodes have to relay the packets coming from their children as well. As a consequence, while some nodes process a single packet, others process a number of packets that is proportional to the number of nodes in the network.



(a) Single line routing topology.



(b) Hierarchical routing topology.

Fig. 5: Histogram of the per-node load in different routing topologies. The store-and-forward and PCAg approaches are compared.

In the PCAg approach, the network load sustained by sensors is proportional to the number q of retained principal components and their number of children in the routing tree. An interesting feature of the PCAg approach is therefore that the network load is more uniformly distributed, and is independent of the network size.

Let us now study under which conditions the adoption of the PCA routing

approach is convenient. From (4) and (5) we derive the following condition on the number q of principal components:

$$\begin{aligned} L^{(\text{pca})} < L &\Leftrightarrow q(C_{\max} + 1) < 2p - 1 \\ &\Leftrightarrow q < \frac{2p - 1}{C_{\max} + 1}. \end{aligned} \tag{6}$$

where $q \leq p$. It follows that if the network size p is sufficiently higher than the topology dependent term C_{\max} , the PCAg strategy outperforms a conventional SF strategy.

The relation (6) is relevant also if we assess the approach in terms of time to first failure (TTFF). The time to first failure is a commonly used metric of network lifetime [7]. It is defined [5, 31] as the elapsed time before a node in the network runs out of energy:

$$\text{TTFF}_{\text{network}} = \min_{i \in V} \{\text{TTFF}_i\}$$

where V is the set of nodes in the network and TTFF_i is the time at which node i runs out of energy. TTTF is dependent on the network load since the radio communication module in a sensor node is the most energy-consuming element (typically at least one order² more consumption than the CPU)

²This ratio is expressed in terms of energy consumption. More specifically, it is worth noting that sending one bit of data consumes as much energy as 2000 CPU cycles, and, therefore, a packet length of 30 bytes (the average packet length in TinyOS) is equivalent to 480000 CPU cycles [30].

[16, 29, 40].

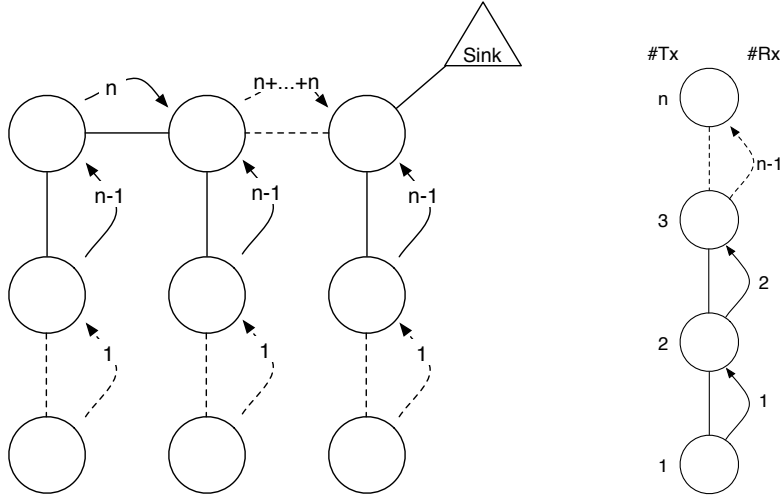
In the store-and-forward approach, each node has to relay an amount of information that depends on its depth in the routing tree. Therefore, TTFF will mostly depend on the lifetime of the nodes *closer* to the sink since these nodes *concentrate* most of the network load.

On the other hand, in a PCAg scenario, each node relays a comparable amount of information (notably the number q of principal components times the number of children C_i). The energy depletion is therefore better distributed in the network and the TTFF does not depend anymore on the size of the network.

In order to better support these preliminary considerations, we detail in the next section the distribution of the network loads on a routing tree inspired from a hierarchical routing topology [35]. We will advocate by means of this particular topology that the overall energy consumption with the PCAg scheme can be reduced by up to an order of magnitude.

4.3 Scalability Analysis

Let us consider the routing topology of Fig. 1 where p sensors are uniformly distributed over a square area of A [m²]. The nodes on a same vertical line are chained together and all chains are connected by means of a single horizontal chain. Moreover, a specific node on the last chain is connected to the data collection sink. If we denote by ρ the density of the sensors (units: [m⁻²]), the total number of nodes is $p = \rho A$ and the side of the grid in Fig.



(a) Amount of packets transferred in the specific case of a hierarchical routing topology.

(b) Summary of the total number of transmitted (Tx) and received (Rx) packets on a single branch of the grid.

Fig. 6: Summary of the network load in a sensor using a hierarchical routing topology and with S/F relaying of the packets.

1 contains \sqrt{p} nodes.

The communication costs can be obtained as follows. Each node has to (i) relay the information originating from the previous nodes on the chain and (ii) send its own measurement. In particular, the first node sends 1 packet, the second one receives 1 packet and sends 2 packets, the third receives 2 packets and sends 3 packets, and so on (Fig. 6(b)). Therefore, along a chain of length $n = \lfloor \sqrt{p} \rfloor$, the total amount of transmitted (Tx) and received (Rx) packets are:

$$\begin{cases} \text{Tx} &= \frac{n(n+1)}{2} \\ \text{Rx} &= \frac{(n-1)n}{2} \end{cases} \quad (7)$$

Furthermore, we can denote by the value \mathcal{E} the average energy cost to transmit or receive a single packet (dimension: [J/pck]). According to [16], these two values are about the same in wireless sensor networks. For instance, typical value for the transmission of 1 bit is of $\mathcal{E} \simeq 20\mu\text{J}$ for the MicaZ board, and of $\mathcal{E} \simeq 50\mu\text{J}$ for the IMote2. Therefore, from (7), we can derive the *order of magnitude* of the relaying cost (in terms of energy) on a single chain, with respect to the length n of this transmission chain:

$$\begin{cases} E_{\text{Tx}} &= \mathcal{E} \frac{n(n+1)}{2} = O(n^2) \\ E_{\text{Rx}} &= \mathcal{E} \frac{(n-1)n}{2} = O(n^2) \end{cases}$$

$$E_{\text{chain}} = E_{\text{Tx}} + E_{\text{Rx}} = O(n^2) + O(n^2) = O(n^2).$$

The same approach applies to the computation of the relaying cost on the horizontal chain in Fig. 6. Each node of this chain receives $k = (n-1)n/2$ packets from the vertical chains. Thus, the first node on the horizontal chain transmits k packets, the following one chain receives the k packets and add its k packets, and so on. Therefore, the cumulative number of packets relayed

on the entire horizontal chain is:

$$\begin{cases} \text{Tx}^{(\text{H})} &= \frac{kn(n+1)}{2} = \frac{n(n+1)}{2} \frac{n(n+1)}{2} = O(n^4) \\ \text{Rx}^{(\text{H})} &= \frac{kn(n-1)}{2} = \frac{n(n+1)}{2} \frac{n(n-1)}{2} = O(n^4) \end{cases}$$

and the order of magnitude of energy cost for the relaying on the entire horizontal chain can be expressed as:

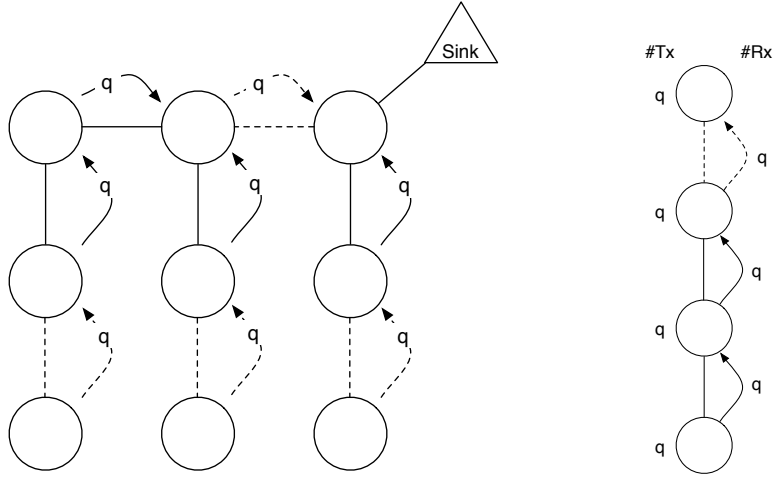
$$E_{\text{horizontal}} = \mathcal{E}\text{Tx}^{(\text{H})} + \mathcal{E}\text{Rx}^{(\text{H})} = O(n^4)$$

The order of magnitude for the energy required to relay the information of the whole network of sensors is:

$$\begin{aligned} E_{\text{network}} &= E_{\text{horizontal}} + nE_{\text{chain}} \\ &= O(n^4) + nO(n^2) \\ &= O(n^4) = O(p^2). \end{aligned} \tag{8}$$

We obtain that the energy required to transport the information using a hierarchical routing in a wireless sensors network increases as the square of the number of nodes.

Let us now analyze the cost for the PCAg scheme. In this case each node sends $q \leq p$ packets per epoch, where p is the total number of nodes. Fig. 7 shows the number of packets transferred on a hierarchical routing topology made of p nodes and Fig. 7(b) details the path of the q components on a



(a) Amount of packets transferred in the specific case of the PCAg compression.

(b) Summary of the total number of transmitted (Tx) and received (Rx) packets on a single chain in an architecture using PCAg compression.

Fig. 7: Summary of the network load in a sensor grid topology and with PCAg relaying of the packets.

specific chain. The number Tx of transmitted packets and the number Rx of received packets along a chain of length n are:

$$\begin{cases} \text{Tx} &= nq \\ \text{Rx} &= (n - 1)q \end{cases}$$

The order of magnitude of energy consumption along a chain of length n is

then:

$$\left\{ \begin{array}{l} E_{\text{Tx}} = \mathcal{E}nq = O(nq) \\ E_{\text{Rx}} = \mathcal{E}(n-1)q = O(nq) \end{array} \right.$$

$$E_{\text{chain}} = E_{\text{Tx}} + E_{\text{Rx}} = O(nq) + O(nq) = O(nq), \quad q \leq n^2$$

On the horizontal chain, each node receives nq components from its corresponding chain. It merges its own sensing information and forwards the packet. This packet is made of nq components and requires nq transmissions. The numbers of transmitted and received packets are:

$$\left\{ \begin{array}{l} \text{Tx}^{(\text{H})} = n(nq) = O(n^2q) \\ \text{Rx}^{(\text{H})} = (n-1)nq = O(n^2q) \end{array} \right.$$

and, in terms of energy:

$$E_{\text{horizontal}} = \mathcal{E}\text{Tx}^{(\text{H})} + \mathcal{E}\text{Rx}^{(\text{H})} = O(n^2q)$$

Finally, the order of magnitude of the whole energy spent to propagate the

values of the sensor by using the PCA compression technique is:

$$\begin{aligned}
E_{\text{network}} &= E_{\text{horizontal}} + nE_{\text{chain}} \\
&= O(n^2q) + nO(nq) \\
&= O(n^2q) = O(pq), \quad q \leq p.
\end{aligned} \tag{9}$$

Equations (8) and (9) show that the adoption of the PCA strategy allows an order of magnitude reduction of the energy consumption.

We can conclude this section by summarizing the added value of the adoption of principal component aggregation scheme in a network architecture: (i) the availability of a traffic control policy which guarantees the maximum of retained information for a given traffic, (ii) an enhanced distribution of the energy depletion, (iii) a significant reduction of the TTFF.

5 Experimental Results

This section illustrates experimentally the different tradeoffs incurred by the principal component aggregation scheme, and compares them to the default S/F scheme. Experiments are based on a set of real-world temperature measurements, which we detail in Section 5.1. Instances of network routing trees are generated according to a simple algorithm described in Section 5.2. Results related to the tradeoffs between approximation errors and network load

are presented in Section 5.3. Section 5.4 then illustrates the ability of the principal component aggregation to better distribute the network load among the sensors.

5.1 Data

Experiments were carried out using a set of five days of temperature readings obtained from a 54 Mica2Dot sensor deployment at the Intel research laboratory at Berkeley [37]. The sensors 5 and 15 were removed as they did not provide any measurement. The readings were originally sampled every thirty-one seconds. A preprocessing stage where data was discretized in thirty second intervals was applied to the dataset. After preprocessing, the dataset contained a trace of 14400 readings from 52 different sensors. The code associated to the preprocessing and the network simulation was developed in R, an open source statistical language, and is available from the authors' website [26].

An example of temperature profile is reported in Fig. 8, and an illustration of the dependency between sensor measurements is given in Fig. 9. The sensors 21 and 49 were the least correlated ones over that time period, with a correlation coefficient of 0.59. They were situated on opposite sides of the laboratory. Temperature over the whole set of data ranged from about 15°C to 35°C.

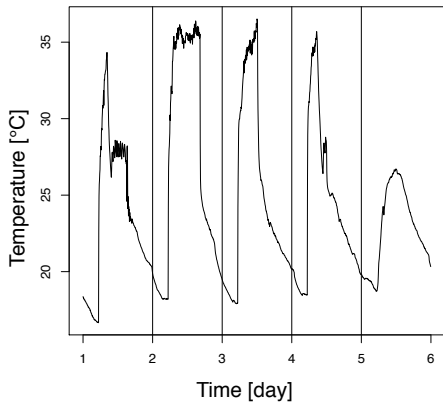


Fig. 8: Temperature measurements collected by sensor 21 over a five day period.

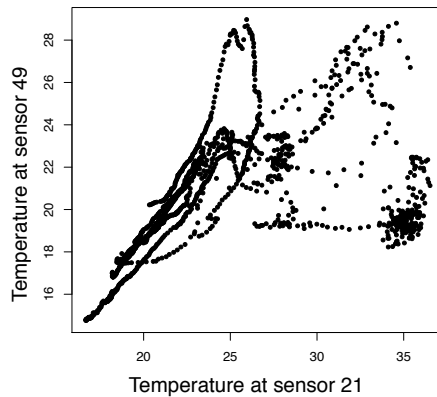


Fig. 9: Examples of the dependencies between the measurements of sensor 21 and sensor 49.

5.2 Network Simulations

The positions of the sensors are provided in [37], and the distribution of the sensors in the laboratory can be seen in Fig. 1. We analyzed the communication costs incurred by different routing trees which were generated in the following way. The root node was always assumed to be the top right sensor node in Fig. 1 (node 16 in [37]). The routing trees were generated on the basis of the sensor positions and the radio range was varied from 6 meters (minimum threshold such that all sensor could find a parent) to 50 meters (all sensors in radio range of the root node). Starting from the root node, sensors were assigned to their parent in the routing tree using a shortest path metric, until all sensors were connected. An illustration of the routing tree obtained for a maximum communication range of 10m is reported in Fig. 1.

5.3 Principal Component Aggregation

The average amount of information retained by the first 25 principal components is reported in Fig. 10. We relied on a cross-validation technique to estimate the amount of variance that could be retained from the dataset. Cross-validation was used to simulate the fact that only part of the data is used to compute the principal components, and was implemented as follows. The dataset was split in ten consecutive blocks (1440 observations – i.e., half a day of measurements). Each of the ten blocks was used as the *training* set to compute the covariance matrix and its eigenvectors, and the remaining observations, referred to as *test* set, were used to estimate the percentage of retained variance.

Fig. 10 provides the average retained variance on the 10 test sets for the first 25 principal components. The upper line gives the amount of variance retained when principal components are computed with the test sets, while for the lower curve the components are computed with the training set. This figure shows that the first principal component accounts on average for almost 80% of the variance, while 90% and 95% of variance are retained with 4 and 10 components, respectively. The confidence level of these estimates (not reported for clarity) was about $\pm 5\%$. Additional experiments, not reported for space constraints, were run using k -cross validation with k ranging from 2 to 30. The percentages of retained variance on the test data blocks tended to decrease with k . Gains of a few percents were observed for k lower than five (more than one day of training data), and losses of a few percents

were observed for k higher than 15 (less than nine hours of data). It should be stressed however that the important point in collecting observations for extracting the principal components is not so much in the number of observations collected, but in their ability to properly capture the covariances between sensor measurements.

The amount of retained variance increases very fast with the first principal component, and becomes almost linear after about ten components. A linear increase of retained variance with the number of principal components reflects the fact that the components obtained by PCA are actually no better than random components [18]. From Fig. 10, it therefore seems that from 10 or 15 components onwards, the remaining variations can be considered as white noise.

Figure 11 illustrates the approximations obtained during the first round of the cross validation (i.e., principal components are computed from the first 12 hours of measurements) for the sensor 49, using one, five and ten principal components. A single principal components provides rough approximations, which cannot account for the specificities of some of the sensor's measurements. For example, the stabilization of the temperature around 20°C around noon during the second, third and fourth day (probably due to the activation of an air conditioning system at a location close to sensor 49) are not rendered by the approximations.

Increasing the number of principal components allows to better approximate the local variations, and passing to five components provides for ex-

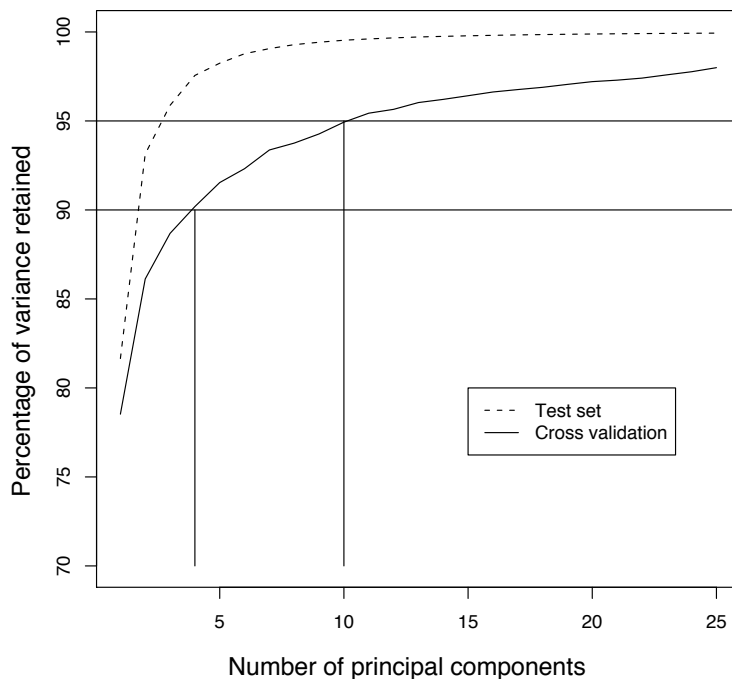


Fig. 10: Minimal amount of principal components required in order to retain a given measurement variance.

ample a much better approximation of the measurements of the sensor 49. Note however that the quality of obtained approximations may not be the same for each sensor. For example, for sensor 22 (results not reported due to space constraints), the gain in approximation accuracy were much higher when passing from five to ten components.

5.4 Communication Costs

We now compare the communication costs incurred by the default and PCAg schemes for different types of routing trees. We illustrate the impact of the

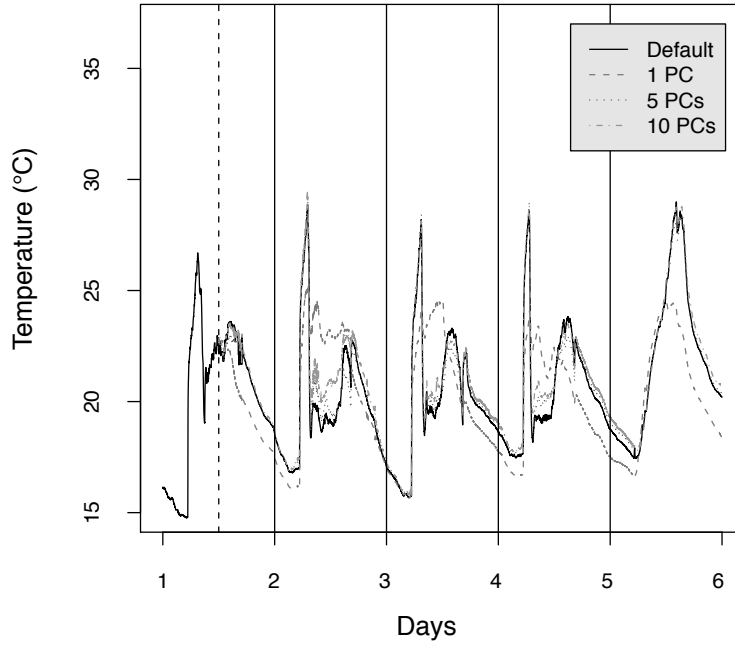


Fig. 11: Approximations obtained on the test set for the sensor 49 using one, five, and ten principal components respectively.

routing tree structure on the number of packets routed in the network by varying the communication range of the radio (see Fig. 12). Given that the sensors choose as their parent the sensor within radio range that is the closest to the base station, increasing the radio communication range leads the routing tree to have a smaller depth, and its nodes to have an average higher number of children.

For the default scheme (Fig. 12, left), increasing the radio range reduces the average sensor network load but does not reduce the maximum network load. The latter is supported by the root node. A radio range of 50 meters allows all nodes to communicate with one another. The resulting routing tree

is of depth one, and contains 51 leaf nodes and one root node. The root node is therefore still required to receive and forward 51 packets, and to transmit its own measurements. The maximum network load therefore remains at 103 packets processed per epoch.

For the PCAg scheme, we first illustrate the communication costs incurred by the computation of one component (Fig. 12, right), and will then generalize after to the costs incurred by the computation of several components. It is interesting to see that increasing the radio range in the PCA scheme tends to increase the network load. This is the opposite effect than the one observed for the default scheme, and is a direct consequence of the increased number of children induced by routing trees with smaller depths. Eventually, for a fully interconnected network, we observe the same effect than for the default scheme, where all sensors send only one packet, while the root node sustains the higher network load due to the forwarding task. Note however that while the root node receives 51 packets, it only has to send one packet thanks to the aggregation process. This therefore bounds the maximum network load to 52 packets per epoch.

The extraction of one component is therefore always beneficial for the network load incurred at the root node, and performances increase as the communication range diminishes. Extracting more components may however be detrimental to the network compared to the S/F scheme. The extraction of k components implies k times the number of receptions and transmissions required for one component. The network load incurred by k components is

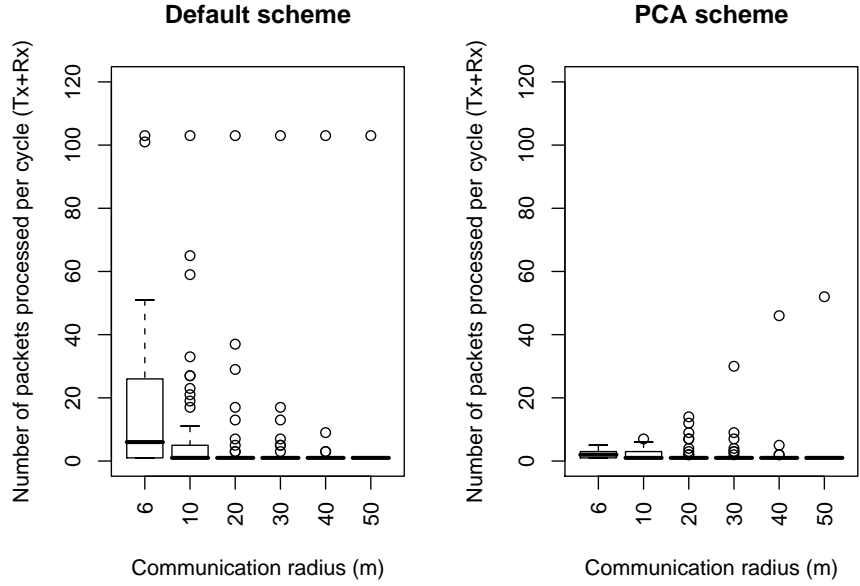


Fig. 12: Communication costs incurred by the default scheme (left) and the aggregation scheme (right), as a function of the radio range.

therefore obtained by multiplying the figures in Fig. 12 by k .

This is illustrated in Fig. 13 where the number of packets processed (received and sent) is reported as a function of the number of principal component extracted for a radio range of 10. In this routing tree, the maximum number of children is 6 (see Fig. 1). For the extraction of one PC, the maximum network load will be therefore be of 7, i.e., 6 receptions and one transmission, to be compared with the maximum network load of 101 for the root node. However, extracting more than 15 components leads the maximum network load to be higher than in the default scheme, as the sensor node aggregating the packets from its 6 children will sustain a network load

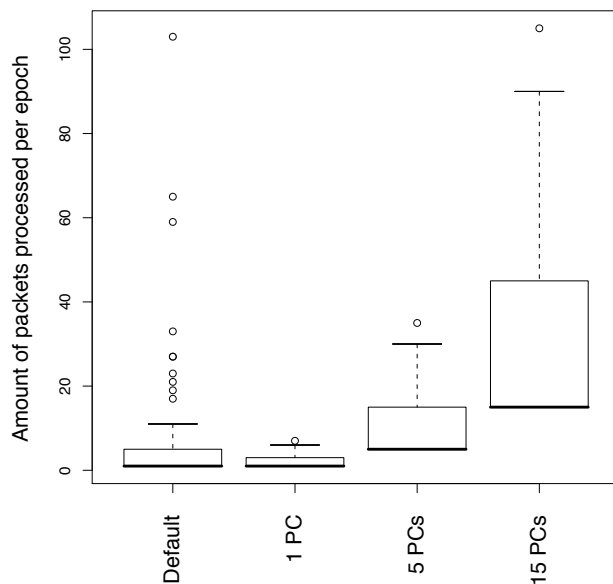


Fig. 13: Comparison of the communication costs incurred by the default scheme, and the aggregation scheme as 1, 5 and 15 principal components are retrieved. Radio range is 10 meters. The maximum network load is reduced if the number of principal components is less than 15.

of 105 packets per epoch.

6 Related Work and Extensions

The application of the PCA to extract features out of wireless sensor data has been suggested at several occasions in the recent literature on data processing techniques for sensor networks. In [22], authors proposed to rely on principal component scores in order to (i) compress vibration sensor data and (ii) to detect events in vibration patterns. Event detection based on principal components has also been addressed in [13], where the authors proposed

to apply the PCA on network statistics, which were considered as sensors' internal state measurements. Their approach was shown to provide a way to detect network anomalies that would not be detected at the sensor scale. PCA has also been suggested as a way to perform event classification in [10], where the application was aimed at classifying vehicles on the basis of vibration sensor data. Finally, the authors in [3] proposed to rely on the PCA as a preprocessing step in a data mining architecture for sensor networks.

Contrary to the scheme presented in this chapter, these approaches rely on clusters of sensors. The computation of the principal component scores is carried out at the cluster level, by means of a coordinating node that gathers measurements in each cluster for computing the scores. The computation is therefore only distributed at the cluster level, whereas the scheme proposed in this chapter allows to distribute this computation at the level of wireless sensors. The proposed scheme and cluster-based approaches are therefore not exclusive, but open the way to hybrid systems that rely on clustered network where the principal component aggregation is used within clusters.

The aggregation principle underlying the compression scheme proposed in this chapter is readily extensible to any basis transformation. Related work on the use of basis change for sensor networks has been addressed in [2, 9], where the authors investigated the use of random bases to project sensor measurements. The work is analyzed in the context of compressed sensing, an emerging field in signal processing (see also [8]). Their work is however mainly focused at the theoretical ability of random bases to retain the sensor

measurements' variations. The possibility of a synchronized routing tree was mentioned, but no further analysis on the tradeoffs between the communication costs and the signal reconstruction accuracy was provided. Among other basis transformations of interest, we stress that the independent component analysis (ICA), also known as blind source separation [15], is particularly appealing. ICA aims at determining a basis which not only decorrelates signals, but that also gets them independent. ICA has for example proven particularly efficient in speech processing for separating the set of independent sources composing an audio signal.

We finally refer the reader to [23, 24] for additional optimization schemes that can be designed to improve the efficiency of aggregation services, particularly in terms of resilience to sensor failure and robustness to missing measurements. In the proposed scheme, missing measurements entail an incomplete computation of the principal component scores, which may lead to unexpectedly poor approximations of the sensor measurements at the base station. As discussed in [23, 24], a common approach for managing missing values is to use inference models based on past observations. More specifically, techniques proposed for sensor networks include *caching* [6, 23, 27], where missing data is simply replaced with the latest observed value. More complex inference models such as autoregressive models [34, 38] or Kalman filters [17] for example have also been proposed, allowing better approximations at the price of higher computational costs.

7 Conclusion

In this chapter we proposed a distributed data compression scheme based on the principal component analysis. This approach, called *principal component aggregation*, allows to evenly distribute among the sensor nodes the computation of the principal component coordinates. The approach was shown to benefit from the two following properties. First, as a by-product of aggregation, the network load is distributed among the sensors and scales with the network size. Second, thanks to the principal component analysis, the linear redundancies between sensor measurements are removed.

A thorough analysis of the tradeoffs involved was conducted, both analytically and experimentally. It showed that, in the case of correlated measurements, the PCA allows to significantly reduce the energy consumption and the network load. Experiments based on real-world temperature measurements illustrated the fact that 90% of the variance of the data could be recovered at the base station while, at the same time, reducing the network load of up to 20%.

8 Acknowledgments

This work was supported by the COMP2SYS project, sponsored by the Human Resources and Mobility program of the European Community (MEST-CT-2004-505079), and by the PIMAN project, supported by the Institute for the Encouragement of Scientific Research and Innovation of Brussels, Bel-

gium. Authors would like to thank Mathieu Van Der Haegen for insightful comments.

9 Biographies

Yann-Aël Le Borgne received his BSc degree in Computer Science from the University of Nantes, France, in 2002, and his MSc in Cognitive Sciences from the University of Grenoble, France, in 2003. In 2004, he was awarded a Marie Curie Research Fellowship from the European Commission, in the context of the Early Stage Research Training Programme COMP2SYS. He is currently PhD student in the Machine Learning Group (MLG) of the Université Libre de Bruxelles, Belgium. His main interests concern the use and development of machine learning techniques for data streams, particularly in the field of Wireless Sensor Networks.

Jean-Michel Dricot received the Master in Engineering degree and the PhD degree in Applied Sciences from the Université Libre de Bruxelles in 2001 and 2007, respectively. He was a postdoc researcher at France Télécom R&D (Orange Labs) and is now with the Machine Learning Group of the Université Libre de Bruxelles. His research interests include the design of routing protocols for wireless sensor networks operating in a strongly faded environment and the cognitive radios architectures (with particular emphasis on the cross-layer cooperation between the physical and the routing layers).

Gianluca Bontempi graduated with honors in Electronic Engineering (Politecnico of Milan, Italy) and obtained his PhD in Applied Sciences (ULB, Brussels, Belgium). He took part to research projects in academy and private companies all over Europe. His interests cover data mining, machine learning, bioinformatics, time series prediction, intelligent control and numerical simulation. He is also co-author of software for data mining and prediction which was awarded in two international competitions. Since 2002 he is associate professor with the Computer Science Department of ULB and head of the ULB Machine Learning Group.

References

- [1] IF Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [2] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk. Distributed compressed sensing. *Submitted for publication*, 2005.
- [3] G. Bontempi and Y. Le Borgne. An adaptive modular approach to the mining of sensor network data. In *Proceedings of the Workshop on Data Mining in Sensor Networks, SIAM SDM*, Newport Beach, CA, April 2005.
- [4] N. Burri and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. *Proceedings of the 6th international conference on*

- Information processing in sensor networks*, pages 450–459, 2007.
- [5] J.H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proc. IEEE INFOCOM 2000*, Tel Aviv, 2000.
- [6] A. Deshpande, S. Nath, P.B. Gibbons, and S. Seshan. Cache-and-query for wide area sensor databases. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 503–514, 2003.
- [7] Qunfeng Dong. Maximizing system lifetime in wireless sensor networks. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 3, Piscataway, NJ, USA, 2005. IEEE Press.
- [8] DL Donoho. Compressed Sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.
- [9] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk. Distributed compressed sensing of jointly sparse signals. In *Proceedings of the 39th Asilomar Conference on Signals, Systems and Computation*, pages 1537–1541. IEEE Signal Processing Society, 2005.
- [10] M.F. Duarte and Y. Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.
- [11] Mark D. Hill. What is scalability? *ACM SIGARCH Computer Architecture News*, 18(4):18–21, 1990.

- [12] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, and N. Taft. In-network PCA and anomaly detection. *Advances in Neural Information Processing Systems, 19th ed.*, B. Scholkopf, J. Platt and T. Hoffman, Ed. Cambridge, MA: MIT Press, 2007.
- [13] L. Huang, X. L. Nguyen, M. Garofalakis, M. Jordan, A. D. Joseph, and N. Taft. In-network PCA and network anomaly detection. In *Advances in Neural Information Processing Systems (NIPS)*, pages 19–. MIT Press, 2006.
- [14] T.T. Huynh and C.S. Hong. A novel hierarchical routing protocol for wireless sensor networks. In *Inter. Conf. on Computational Science and its Applications, ICCSA 2005*, pages 339–347, 2005.
- [15] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. J. Wiley New York, 2001.
- [16] M. Ilyas, I. Mahgoub, and L. Kelly. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, Inc. Boca Raton, FL, USA, 2004.
- [17] A. Jain and E.Y. Chang. Adaptive sampling for sensor networks. *ACM International Conference Proceeding Series*, pages 10–16, 2004.
- [18] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.
- [19] P. R. Kumar and P. Gupta. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

- [20] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. *SIGCOMM Comput. Commun. Rev.*, 34(4):219–230, October 2004.
- [21] Y. Le Borgne and G. Bontempi. Unsupervised and supervised compression with principal component analysis in wireless sensor networks. In *Proceedings of the Workshop on Knowledge Discovery from Data, 13th ACM International Conference on Knowledge Discovery and Data Mining*, pages 94–103, New York, NY, 2007. ACM Press.
- [22] J. Li and Y. Zhang. Interactive sensor network data retrieval and management using principal components analysis transform. *Smart Materials and Structures*, 15:1747–1757(11), December 2006.
- [23] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. *Proceedings of the ACM Symposium on Operating System Design and Implementation (OSDI)*, 2002.
- [24] S.R. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems (TODS)*, 30(1):122–173, 2005.
- [25] K.V. Mardia, J.T. Kent, J.M. Bibby, et al. *Multivariate analysis*. Academic Press New York, 1979.

- [26] Wireless Sensor Lab. Machine Learning Group. University of Brussels. <http://www.ulb.ac.be/di/labo/index.html>.
- [27] C. Olston, B.T. Loo, and J. Widom. Adaptive precision setting for cached approximate values. *ACM SIGMOD Record*, 30(2):355–366, 2001.
- [28] S. Patten, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *Proceedings of the third international symposium on Information processing in sensor networks*, pages 28–35, 2004.
- [29] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364–369, 2005.
- [30] V.S. Raghunathan and C.S.P. Srivastava. Energy-Aware Wireless Microsensor Networks. *Signal Processing Magazine, IEEE*, 19(2):40–50, 2002.
- [31] A. Sankar and Z. Liu. Maximum lifetime routing in wireless ad-hoc networks. In *Proc. INFOCOM'04*.
- [32] A. Scaglione and S. Servetto. On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks. *Wireless Networks*, 11(1):149–160, 2005.

- [33] TinyOS. Project Website: <http://www.tinyos.net>.
- [34] D. Tulone and S. Madden. PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks. *Wireless Sensor Networks: Third European Workshop, EWSN 2006, Zurich, Switzerland, February 13-15, 2006: Proceedings*, 2006.
- [35] Maneesh Varshney and Rajive Bagrodia. Detailed models for sensor network simulations and their impact on network performance. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 70–77, New York, NY, USA, 2004. ACM.
- [36] Yong Wang, Chieh-Yih Wan, Margaret Martonosi, and Li-Shiuan Peh. Transport layer approaches for improving idle energy in challenged sensor networks. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 253–260, New York, NY, USA, 2006. ACM.
- [37] Intel Lab Data webpage. <http://db.csail.mit.edu/labdata/labdata.html>.
- [38] S. Santini Y. Le Borgne and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Journal of Signal Processing (Elsevier pub.)*, 87(12):3010–3020, December 2007.
- [39] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(3):9–18, 2002.

- [40] F. Zhao and L.J. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004.