Sensor fusion and estimation strategies for data traffic reduction in rooted wireless sensor networks

A. Agnoli, A. Chiuso, P. D’Errico, A. Pegoraro, L. Schenato

Abstract—One of the main issues with wireless sensor networks (WSNs) is communication efficiency to reduce power consumption. This is often achieved through data routing with aggregation. However, aggregation does not consider possible measurements correlation. This a priori information can be used for data fusion, in order to remove correlation of transmitted data, thus reducing amount of information to be transmitted. In this paper we explore different information processing strategy for state estimation of dynamic linear systems in the context of rooted wireless sensor networks. We propose three data fusion methods: standard measurement aggregation, measurement fusion, and fusion of partial state estimates. These strategies are specifically designed to include possible delayed or dropped packets. Finally, they are compared in term of power consumption efficiency, estimation accuracy, computational and memory complexity, showing tradeoffs between these metrics.

I. INTRODUCTION

Recent advances in low-power analog and digital electronics have enabled mass production of small sensor nodes with sensing, computation, and communication capabilities. This has spurred a substantial amount of research on wireless sensor networks (WSNs) over the past few years. For ease of deployment, sensor devices should be inexpensive, small, and have a long lifetime, which requires the development of very efficient software and hardware solutions. For this reason, protocols for sensor networks should be carefully designed so as to make the most efficient use of the limited resources in terms of energy, computation, and storage.

In this paper we focus on an important aspect of sensor networks, namely in-network data aggregation and fusion. These techniques allow to trade off communication and computational complexity for estimation performance. The basic principle is that local computation often consumes significantly less energy than communication. In typical sensor network scenarios, data is collected by sensor nodes throughout some area, and needs to be made available at some central sink node(s), where it is processed, analyzed, and then used for some decision making process. In many cases, data generated by different sensors can be jointly processed while being forwarded towards the sink, e.g., by fusing together sensor readings related to the same event or physical quantity, or by locally processing raw data before this is transmitted. In-network aggregation deals with this distributed processing of data within the network. Data aggregation techniques are tightly coupled with how data is gathered at the sensor nodes as well as how packets are routed through the network, and have a significant impact on energy consumption and overall network efficiency (e.g., by reducing the number of transmissions or the length of the packets to be transmitted). Also, we emphasize that data size reduction through in-network processing shall preserve as much as possible statistical information about the monitored event.

II. PREVIOUS WORK

In-network signal processing and, more specifically, distributed estimation and sensor fusion have been widely studied since the late seventies in many different areas including control, signal processing, economics and communications. Therefore, the literature on the subject is rather vast, and here we shall only mention the most important results and current trends. In particular, as mentioned above, here we focus on hierarchical dynamic estimation on multi-hop communication trees, as shown in Figure 1. This scenario is different from another popular area of investigation where estimation need to performed at multiple locations such as at each sensor location rather than in one centralized location [1][2][3]. In particular, in that framework it is shown how fast local communication among sensors in between sampling times can be used to retrieve the sufficient statistics for optimal state estimation [4][5]. However, such approach is expensive in terms of exchanged messages while in this work we are interested in strategies that reduce communication as much as possible. More relevant to this work, is the study of how to compute the sufficient statistics to reconstruct the optimal centralized estimate, i.e. the best estimate obtained if all measurements were available at a central location, from distributed processing. In particular, [6][7][8][9][10] show how it is possible to reconstruct the centralized estimate from local estimates computed by each sensor in a scenario where measurement noises among sensors are uncorrelated. However, their approach assumes that the sensors can transmit their local estimate directly to the central node with no delay or packet loss. Finally, there is also considerable work related to sensor fusion in the presence of random delay [11][12] in the measurement, which does not require the buffering of past measurements.

The contribution of this paper is to propose and analyze different estimation strategies for rooted wireless sensor

A. Agnoli, P. D’Errico, A. Pegoraro, L. Schenato are with Department of Information Engineering, University of Padova, Italy schenato@dei.unipd.it
A. Chiuso is with the Department of Management and Engineering, University of Padova, Vicenza, Italy. (email: chiuso@dei.unipd.it)
This work has been partially supported by European Union project SENSNET founded by Marie Curie IRG grant n.014815 and by CARIPARO Foundation.
In this section, we present the most natural approach to perform the state estimation. This strategy consists in simply aggregating raw measurements along the tree and then computing the state estimate at the base station. More precisely, every sensor node $i$ at each sampling time $t$ takes a measurement $y_t$, and then sends a packet to its parent consisting of the packet header, the measurement $y_t$, the depth of the $i$-th sensor node from the root, and with $d_m = \max_i d_i$ the depth of the tree. We assume that there is no delay nor packet loss between the root node and the base station. Finally, we assume that the communication tree topology changes only rarely so that a constant tree can be safely assumed and it is known to the base station. Finally, we assume that each sensor node has some computational and memory resources to implement some local signal processing, and that they are all synchronized, i.e. they share a common notion of time. This modeling is a good representation of many routing protocols for WSNs for environmental monitoring.

IV. MEASUREMENT AGGREGATION

In this section, we present the most natural approach to perform the state estimation. This strategy consists in simply aggregating raw measurements along the tree and then computing the state estimate at the base station. More precisely, every sensor node $i$ at each sampling time $t$ takes a measurement $y_t$, and then sends a packet to its parent consisting of the packet header, the measurement $y_t$, the depth of the $i$-th sensor node from the root, and with $d_m = \max_i d_i$ the depth of the tree. We assume that there is no delay nor packet loss between the root node and the base station. Finally, we assume that the communication tree topology changes only rarely so that a constant tree can be safely assumed and it is known to the base station. Finally, we assume that each sensor node has some computational and memory resources to implement some local signal processing, and that they are all synchronized, i.e. they share a common notion of time. This modeling is a good representation of many routing protocols for WSNs for environmental monitoring.

IV. MEASUREMENT AGGREGATION

In this section, we present the most natural approach to perform the state estimation. This strategy consists in simply aggregating raw measurements along the tree and then computing the state estimate at the base station. More precisely, every sensor node $i$ at each sampling time $t$ takes a measurement $y_t$, and then sends a packet to its parent consisting of the packet header, the measurement $y_t$, the depth of the $i$-th sensor node from the root, and with $d_m = \max_i d_i$ the depth of the tree. We assume that there is no delay nor packet loss between the root node and the base station. Finally, we assume that the communication tree topology changes only rarely so that a constant tree can be safely assumed and it is known to the base station. Finally, we assume that each sensor node has some computational and memory resources to implement some local signal processing, and that they are all synchronized, i.e. they share a common notion of time. This modeling is a good representation of many routing protocols for WSNs for environmental monitoring.

IV. MEASUREMENT AGGREGATION

In this section, we present the most natural approach to perform the state estimation. This strategy consists in simply aggregating raw measurements along the tree and then computing the state estimate at the base station. More precisely, every sensor node $i$ at each sampling time $t$ takes a measurement $y_t$, and then sends a packet to its parent consisting of the packet header, the measurement $y_t$, the depth of the $i$-th sensor node from the root, and with $d_m = \max_i d_i$ the depth of the tree. We assume that there is no delay nor packet loss between the root node and the base station. Finally, we assume that the communication tree topology changes only rarely so that a constant tree can be safely assumed and it is known to the base station. Finally, we assume that each sensor node has some computational and memory resources to implement some local signal processing, and that they are all synchronized, i.e. they share a common notion of time. This modeling is a good representation of many routing protocols for WSNs for environmental monitoring.
In other words, this simply indicates that each row \(i\) of this buffer corresponds to the measurements received from \(i\)-th sensors, and that each column \(h\) corresponds to the measurements taken at time \(t - h\), where \(t\) is the current time, i.e. the base station stores only the received measurements which have a delay smaller or equal to \(d_m\). If the measurement corresponding to the \((i, h)\)-th slot has not been received, then a zero is stored, as shown in Figure 3.

![Fig. 3. Memory data structure required at the base station for the measurement aggregation strategy.](image-url)

It has been shown in [13] that the minimum variance state estimator computable at the base station based on the communication model above, i.e. 

\[
\hat{x}_{t|t} = E[x_t|\hat{y}_{t,0}, \ldots, \hat{y}_{t,t-1}, \gamma_t, 0, \ldots, 0, \gamma_t, 1, \ldots, \gamma_t, T],
\]

where \(\hat{y}_{t,h} = (\hat{y}_{t,0}, \ldots, \hat{y}_{t,h})\) and \(\gamma_{t,h} = (\gamma_{t,1}, \ldots, \gamma_{t,h})\), is given by a time-varying Kalman filter with buffered measurements. More precisely, let us define \(\hat{x}_{t|k}^h = E[x_k|\hat{y}_{t,0}, \ldots, \hat{y}_{t,h}, \gamma_t, 0, \ldots, \gamma_t, T]\), and the corresponding error covariance \(P_{t|k}^h = E[(x_k - \hat{x}_{t|k}^h)(x_k - \hat{x}_{t|k}^h)^T|\hat{y}_{t,0}, \ldots, \hat{y}_{t,h}, \gamma_t, 0, \ldots, \gamma_t, T]\) then \(\hat{x}_{t|t} = \hat{x}_{t|t-1}^t\), where \(\hat{x}_{t|t-1}^t\) is iteratively computed as follows:

\[
\hat{x}_{t-d_m-1:t-d_m-1} = \hat{x}_{t-d_m-1:t-d_m-1}^{t-1}, P_{t-d_m:t-d_m} = P_{t-d_m:t-d_m}^{t-1}
\]

\[
\hat{x}_{t-k+1|k} = A \hat{x}_{t-k|k}, \quad k = t-d_m, \ldots, t-1, t
\]

\[
P_{t-k+1|k} = AP_{t-k|k}A^T + Q
\]

\[
\hat{x}_{t-k|k} = \hat{x}_{t-k|k-1} + K_{t,k}(\hat{y}_{t,k} - C_{t,k} \hat{x}_{t-k|k-1})
\]

\[
P_{t-k|k} = P_{t-k|k-1} - P_{t-k|k-1}C_{t,k}^T(S_{t,k} + R_{t,k})^{-1}C_{t,k}P_{t-k|k-1}
\]

\[
K_{t,k} = P_{t-k|k-1}C_{t,k}(S_{t,k} + R_{t,k})^{-1}
\]

where \(()^T\) indicates the pseudoinverse operator, \(C_{t,k}^T = [\gamma_{t,k}^1 C_{t,k}^T \gamma_{t,k}^2 C_{t,k}^T \ldots \gamma_{t,k}^N C_{t,k}^T]\), and the matrix \(R_{t,k} \in \mathbb{R}^{n \times n}\) is such that the \((i,j)\)-th element is \(R_{t,k}^{ij} = \gamma_{t,k}^i \gamma_{t,k}^j \gamma_{Rij}\). In other words \(C_{t,k}\) and \(R_{t,k}\) can be obtained from the matrices \(C\) and \(R\) by simply replacing the rows and columns of the lost measurements with zeros. Alternatively, one could use the same equations above by constructing the vector \(\hat{y}_{t,k}\) using only the measurements that have arrived, and constructing the matrices \(C_{t,k}\) and \(R_{t,k}\) by removing from \(C\) and \(R\) the rows and columns of the lost measurements, respectively. In this latter formulation, if no measurement from sampling time \(k\) has arrived yet at time \(t\), we set \(C_{t,k} = 0, R_{t,k} = 0, K_{t,k} = 0\).

Informally speaking, the optimal estimator provided above is obtained by iterating at every time step \(t\) a standard Kalman Filter for \(d_m+1\) times using only the \(d_m\)-most recent measurements that have arrived. Equation (3) indicates the variables that needs to be stored to properly initialize the Kalman filter at each time step \(t\). This strategy requires the storage of a matrix of dimension \(n\), and of \(d_m+2\) vectors of dimension \(m\), i.e. it has memory complexity \(O(n + d_m m)\). It also requires the inversion of possibly \(d_m + 1\) matrices of dimension \(m\) at any time \(t\), therefore it has complexity of approximately \(O(d_m m^3)\). Although this complexity can grow large for WSNs with hundreds of sensor nodes, this might not be a problem since, in general, the base station has large computational and memory resources. We will come back to the complexity issues in the conclusions.

Note that this strategy does not require a tree-based communication topology, therefore it is suitable also for communication protocols that send the same packet along multiple routes to reduce packet loss at the price of larger communication load. The only key assumptions here are that the base station is able to sort measurements according to their sampling time and their sensor node origin, and that there is a maximum delay for any packet that successfully arrives.

V. MEASUREMENT FUSION

In order to remove data redundancy, data fusion techniques are to be used, instead of simple measurement aggregation. The problem is now to find a relatively simple fusion function, that can reduce the communication volume without dropping information on the observed process. In particular, in this section we exploit a method based on the Kalman filter equations written in their information form. The information form of the Kalman filter, also called inverse covariance filter, is well known in the literature [14] and it is obtained by applying the Matrix Inversion Lemma to Equations (6)-(8) which can be written as:

\[
\hat{x}_{t|k} = P_{t|k}^*(\hat{x}_{t|k-1}^* + C_{t,k}^T R_{t,k} \hat{y}_{t,k})
\]

\[
P_{t|k}^* = ((P_{t|k-1}^*)^{-1} + C_{t,k}^T R_{t,k} C_{t,k})^{-1}
\]

This new representation of the Kalman filter is useful if the measurement noises are uncorrelated, i.e. if \(R_{ij} = 0, i \neq j\) or equivalently \(R = \text{diag}(R_{11}, R_{22}, \ldots, R_{NN})\). Under this
additional assumption the previous equations can be rewritten as
\[
\hat{x}_{k|k} = P_{k|k}^{i} \left( P_{k|k-1}^{i} \right)^{-1} \hat{x}_{k|k-1}^{i} + \sum_{i=1}^{N} \tilde{\gamma}_{t,k}^{i} C_{t} R_{t}^{-1} y_{t,k}^{i} \quad (9)
\]
\[
P_{k|k}^{i} = \left( (P_{k|k-1}^{i})^{-1} + \sum_{i=1}^{N} \tilde{\gamma}_{t,k}^{i} C_{t} R_{t}^{-1} C_{t}^{T} \right)^{-1} \quad (10)
\]
The terms in the summation of Equation (9) are known as the information vectors and can be computed locally by each node as follows:
\[
z_{k}^{i} = C_{t}^{T} R_{t}^{-1} y_{t,k}^{i}
\]
Note that \(z_{k}^{i} \in \mathbb{R}^{n}\) has the same dimension of the state space. Since Equation (9) requires only the sum of all information vectors \(z_{k}^{i}\), then the sensor nodes do not need to forward them to the parent node similarly to the raw measurement in the previous section, but they can sum the cumulative information vector \(\xi_{t}^{i}\) they have received from their children and pass it on. The only care it needs to be taken is to sum only information vectors corresponding to the same sampling time. More formally each node compute the cumulative information vectors as follows:
\[
\xi_{t}^{i} = \sum_{j \in C(i)} v_{t,k}^{j} \xi_{k}^{j}, \quad k = t - d_{m}, \ldots, t - 1 \quad (11)
\]
where \(v_{t,k}^{j}\) is a binary local variable that is equal to zero if the packet from the \(j\)-th sensor node at time \(t\) was lost, and it is equal to one otherwise. Note that if \(v_{t,k}^{j} = 0\), then all the \(\xi_{k}^{j}\) vectors coming from the \(j\)-branch are lost. In other words, node to the base station can be used to compute the sufficient statistics required by Equation (9). In fact we have:
\[
\zeta_{t,k}^{i} = \sum_{i=1}^{N} \tilde{\gamma}_{t,k}^{i} C_{t} R_{t}^{-1} y_{t,k}^{i}, \quad \zeta_{t,k} = \sum_{i=1}^{N} \gamma_{t,k}^{i} C_{t} R_{t}^{-1} y_{t,k}^{i}
\]
where \(k = t - d_{m}, \ldots, t - 1\). The identities of the sensor nodes which contributed to \(\zeta_{t,k}\) are used to compute \(\sum_{i=1}^{N} \gamma_{t,k}^{i} C_{t}^{T} R_{t}^{-1} C_{t}\) necessary for Equation (10). Figure 5 shows the data structure required at the base station.

![Fig. 5. Memory data structure required at the base station for the measurement fusion strategy.](image)

The state estimate obtained at the base station \(\hat{x}_{t|t}^{BS}\) using this strategy is exactly the same as the one obtained with the scheme in the previous section, but requires less computation. In fact the memory required at the base station is \(O(d_{m} n + n^2)\) necessary to store the matrix \(P_{t|t-d_{m}|t-d_{m}}^{i}\) and the \(d_{m}\) vectors \(\zeta_{t,k}\), while the computational complexity is \(O(d_{m} n^3)\) necessary to invert \(2d_{m}\) matrices of dimension \(n\).

**VI. FUSION OF PARTIAL ESTIMATES**

In this Section we consider a different strategy for reducing the communication load. The computation of the state estimate \(\hat{x}_{k|k}\) will be distributed across the network and only certain “partial” state estimates, which will be formally defined later on, will be transmitted through the network.

First of all we consider an augmented state space which allows to handle very easily the communications delays. Let \(d_{m}\) the maximum number of delays (i.e. the depth of the communication tree). Define by \(\eta_{k} := [x_{k}^{T}, x_{k-1}^{T}, \ldots, x_{k-d_{m}}^{T}]^{T}\) the augmented state. Assume also that the communication center receives the measurements \(y_{k}^{i}\) of the \(i\)-th node with \(d_{i}\) delays. If we denote with \(\bar{y}_{k}^{i}\) the data available to the fusion center from node \(i\) at time \(k\), clearly \(\bar{y}_{k}^{i} = y_{k-d_{i}}^{i}\) holds true. Therefore, an estimator which accounts for the communications delays can be designed based on the augmented model:
\[
\eta_{k+1}^{+} = A \eta_{k} + w_{k}, \quad \bar{y}_{k}^{i} = \bar{C} \eta_{k} + v_{k}^{i} = C_{i} x_{k-d_{i}}^{i} + v_{k-d_{i}}^{i}, \quad i = 1, \ldots, n
\]
where
\[
A := \begin{bmatrix}
A & 0 & \cdots & 0 \\
I & 0 & \cdots & 0 \\
& \ddots & \ddots & \ddots \\
0 & \cdots & I & 0
\end{bmatrix}, \quad w_{k} := \begin{bmatrix}
w_{k} \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

1We shall see that, in fact, measurements are not sent through the network.
In order to fix notation let us denote with \( \hat{\eta}_{k|k} \) the estimator of the augmented state \( \eta_k \) using data up to time \( k \) and consider the (steady state) "centralized" Kalman filter:

\[
\begin{align*}
\hat{\eta}_{k|k} &= F_{k|k} \hat{\eta}_{k-1|k} + L_k \bar{y}_k \\
\bar{y}_k &= C_{\eta} \hat{\eta}_{k|k} + v_k \\
\hat{\eta}_{k|k} &= \sum_{i=1}^{n} \bar{y}_{i|k} 
\end{align*}
\]

where \( L := [L_1, \ldots, L_n] = APCT (CPCT + R)^{-1} \), \( P \) is the (augmented) state prediction error covariance which is the solution of the standard Algebraic Riccati Equation \([14]\), and \( F = A - LC \).

Inspired by \([10]\), the computations required in order to obtain the state estimate \( \hat{\eta}_{k|k} \) can be distributed across the network as follows:

\[
\begin{align*}
\hat{\eta}_{i|k} &= F_{i|k} \hat{\eta}_{i-1|k} + L_i \bar{y}_k \\
\bar{y}_k &= C_{\eta} \hat{\eta}_{k|k} + v_k \\
\hat{\eta}_{k|k} &= \sum_{i=1}^{n} \bar{y}_{i|k}
\end{align*}
\]

where the \( i \)-th node is responsible for computing \( \hat{\eta}_{i|k} \), which we shall call a "partial" estimate of the state \( \eta_k \). Note that \( \hat{\eta}_{i|k} \) uses only \( \bar{y}_k = y_k - d_i \). Indeed one can think that, at time \( k - d_i \), node \( i \) computes \( \hat{\eta}_{i|k} \) which, sent through the network, reaches the fusion center at time \( k \). Of course, when estimates are sent to the fusion center through some other node, the parent node can fuse its "partial" estimate with those coming from its children just summing the partial estimates. This idea is described pictorially in Figure 6.

Note that the amount of information to be sent through the network depends on the augmented state dimension, which is \( nd_m \), therefore it can be very large for deep trees. However, it turns out that very standard model reduction techniques can be used to the purpose of designing low dimensional filters/estimators. Ideally \( L_2 \) weighted model reduction would allow to find the optimal (in the sense of minimum variance) approximation of the optimal filter (see e.g. \([15]\)). Unfortunately \( L_2 \) model reduction turns out to be rather difficult in general and several alternatives are possible, ranging from the classical balanced model reduction to optimal Hankel norm model reduction as suggested for instance in \([15]\). In our experience, a reduced model of order \( n \) (i.e. of the same order as the original state) yields very good performances close to the optimal, full order filter. Using a reduced order filter the computational and memory complexity at the base station is \( O(1) \) and \( O(n) \), respectively, while computational and memory complexity at each sensor node is \( O(n^2) \) and \( O(n^2) \), respectively. Therefore, this strategy distributes much of the computational and memory burden from the base station to the sensor nodes.

It is rather simple to see that this algorithm provides indeed the optimal estimator, similarly to other fusion techniques presented earlier, in the absence of packet losses. Unfortunately, the situation is radically different in the presence of packet losses. When packets are lost, the fusion center does not receive some of the partial estimators \( \hat{\eta}_{i|k} \). In this scenario, it is not clear how these should be replaced in the computation of the estimator

\[
\hat{\eta}_{k|k} = \sum_{i=1}^{n} \hat{\eta}_{i|k}.
\]

At a first glance letting the \( i \)-th partial estimator to evolve in open loop, i.e. setting \( \hat{\eta}_{k|k} = A_{\eta} \hat{\eta}_{k-1|k-1} \) when the \( i \)-th partial estimate is lost, could be a sensible thing to do. Unfortunately it can be proved that this is not the "right" thing to do. We have indeed verified in a number of simple examples, that this strategy does not work and yields very bad estimates even in the presence of very few packet losses.

For the purpose of comparison we have also implemented the decentralized fusion strategy suggested in \([8]\), adapted to our setup. It turns out that also this latter algorithm, while providing the optimal estimator in the absence of packet losses, fails when packets are lost. Therefore extending these schemes also in the presence of packet losses remains an open research question which we are planning to address in our future work.

As a last remark let us observe that differently from the strategy presented in this section, the strategies presented earlier (i.e. measurements aggregation and measurements fusion) do work also in the presence of packet losses and are, to the best of our knowledge, the approaches which perform better in this situation.

**VII. Power consumption considerations**

Another important parameter to be evaluated is the energy consumption of the estimation strategies presented above. In particular, energy consumption is directly proportional to the number of bits transmitted. Here we compute the average number of bits transmitted by each sensor node at any time instant. This number depends on the specific topology of the WSNs tree considered. To derive some useful guidelines, we consider complete \( \ell \)-ary trees, i.e. trees where each node has \( \ell \) children, of depth \( d_m \), where the root node has depth zero. We define with the variables \( b_h, b_m, b_t, b_v \) the number of bytes necessary to encode the packet header, the sensor node ID,
address, the sampling time, and a real number, respectively. We also assume that each sensor measurement has dimension \( m \), while the dimension of the state is \( n \). It is useful to define the following quantities:

\[
M_1 = \sum_{i=0}^{d_m} k^i, \quad M_2 = \sum_{i=0}^{d_m} k^i(i+1), \quad M_3 = \sum_{i=0}^{d_m} k^i(d_m - i + 1)
\]

Note that \( M_1 \) corresponds also to the total number of sensor nodes in the WSN, i.e. \( M_1 = N \). The total number of bits transmitted at each time step by the whole network for each of the three strategies presented are given by

\[
\begin{align*}
N_{ma} & = M_1 b_h + (m b_r + b_l + b_a) M_2 \\
N_{mf} & = M_1 b_h + b_o M_2 + (n b_r + b_l) M_3 \\
N_{sf} & = M_1 (b_h + n b_r)
\end{align*}
\]

where \( N_{ma}, N_{mf}, N_{sf} \) stand for measurement aggregation, measurement fusion, and partial state estimate fusion, respectively. It is clear that from the previous equations that \( N_{sf} < N_{mf} \), i.e. the partial state estimate fusion strategy requires less energy then the measurement fusion. Differently, if \( n > m \) then for shallow trees it is more convenient to use the measurement aggregation strategy, while for deeper trees the state fusion and then the measurement aggregation strategies are to be preferred, as graphically illustrated in Figure 7, where we considered the following values: \( k = 3, b_h = 4, b_o = 2, b_l = 2, b_a = 6 \). In fact, in this example, loss and delay. We showed how they all exhibit different tradeoffs between estimation accuracy, computational and memory requirement at the base station and at the sensor nodes, and power consumption. These tradeoffs are also strongly dependent on the tree topology, i.e. if it is shallow of deep, and on the ratio between the measurement dimension and the state dimension. Therefore it is not possible to elect a best strategy in absolute terms, but it needs to be selected based on the specific application in mind.

Probably one of the most important findings of this paper was to show that traditional distributed estimation strategies [8][10] cannot be easily extended to cope with packet losses. Indeed packet loss, which is an unavoidable feature of WSNs, opens a rather unexplored avenue of research for distributed estimation and signal processing in WSNs.

Finally, even for the measurement aggregation and the measurement fusion strategies, which provide good performance also under lossy communication, there is space for improvement in particular in terms of computational complexity at the base station.

**REFERENCES**


![Fig. 7. Average number of bytes transmitted per node per time step as a function of the tree depth d. On the x-axis inside the round parentheses there are also indicated the corresponding total number of sensor nodes in the network.](image-url)