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.

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.

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 currents 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

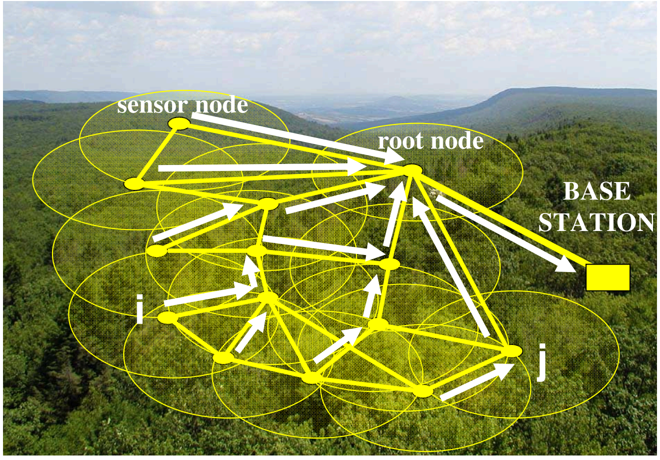

Fig. 1. Pictorial representation of Wireless Sensor Network for environment monitoring. The small dots indicate the location of the sensing nodes, the shaded circles indicate the sensing regions and the segments the communication links. White arrows represent the tree-based routing paths from the sensor nodes to the root sensor node and to the base station.

networks where communication messages are subject to delay and packet loss. In particular we want to compare them in terms of estimation accuracy, computational and memory complexity required by each sensor node and by the base station, and energy consumption. Although optimal distributed estimation strategies have been proposed [8][10], they cannot be easily extended to a scenario which includes packet loss and delay, and very little work has been done with this respect. Also, we show that the optimal strategy in term of power consumption strongly depends on the communication topology, since for shallow trees it is more convenient to send raw measurements, while for deeper trees it is better to locally fuse information.

III. Problem formulation

We consider a discrete time linear stochastic systems observed by N sensors:

$$x_{t+1} = Ax_t + w_t \quad (1)$$

$$y_t^i = C_i x_t + v_t^i, \quad i = 1, \dots, N \quad (2)$$

where $x \in \mathbb{R}^n$, $y_i \in \mathbb{R}^{m_i}$, w_t and v_t^i are gaussian noises with zero mean and such that $\mathbb{E}[w_t w_s^T] = Q\delta(t-s)$, $\mathbb{E}[v_t^i w_s^T] = 0, \forall i$, and $\mathbb{E}[v_t^i (v_s^j)^T] = R_{ij}\delta(t-s)$, i.e. we also allow for correlated measurement noise. More compactly, if we define the compound measurement noise vector $v_t = (v_t^1, \dots, v_t^N) \in \mathbb{R}^m, m = \sum_i m_i$, we have $\mathbb{E}[v_t v_s^T] = R\delta(t-s)$, where the the (i,j)-th block of the matrix $R \in \mathbb{R}^{m \times m}$ is $[R]_{ij} = R_{ij} \in \mathbb{R}^{m_i \times m_j}$. We also assume that $R > 0$, the pair $(A, Q^{1/2})$ is reachable and (A, C) is detectable, where $C^T = [C_1^T \; C_2^T \; \dots \; C_N^T]$, which are necessary conditions for the existence of a stable estimator.

The sensors are not physically co-located and can communicate only with their neighbors, thus forming a communication graph, as shown in Figure 1. From this graph a special node is elected as data collector, and from this node a rooted tree consistent with the communication graph is then constructed. Therefore, each sensor in the network also acts as a router for the messages from its children sensors to its parent sensor, till all data are collected at the root. The root then transmits the gathered information to a base station where the state estimation is required. Without loss of generality, we assume that the root sensor is sensor y^1. We also indicate with $\mathcal{C}(i)$ the children of the i-th sensor node. Any transmission from a children node to a parent node is subject packet loss, and there is a unitary time step delay if it is successful, therefore data from any node can arrive to the root node with a time delay which is no smaller then its depth in the communication tree. We indicate with d_i 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. We finally 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 startegy 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 time t it was taken, and its identification number i. If node i has some children, it simply aggregates the messages it has received from them into the packet to be sent to the parent as pictorially illustrated in Figure 2. Using this communication scheme, the sensor nodes closer to the root need to send large packets since they have to transmit not only their measurement but also all the measurements collected by the sensor nodes below them. On the other hand, using this strategy there is no need for any in-network signal processing.

Measurements from all sensor nodes arrive at the base station with random delay or might even be lost along the way. Let us indicate with τ_k^i the total delay of the measurement y_k^i when received at the base station, therefore, according to our modeling the packet either has been correctly delivered and its delay corresponds to the depth of i-th sensor, i.e. $\tau_k^i = d_i$, or it has been lost along the way, i.e. $\tau_k^i = \infty$. Since the maximum delay of any measurement correctly delivered at the base station is given by d_m, we can simply store all recently received measurements in a buffer of $N \times (d_m + 1)$ elements, where the (i,h)-th entry $(i \in \{1,..,N\}, h \in \{0,1,..,d_m\})$ is given by $\tilde{y}_{t,t-h}^i = \gamma_{t,t-h}^i y_{t-h}^i$, and the auxiliary variable $\gamma_{t,t-h}^i$ is defined as

$$\gamma_{t,t-h}^i = \begin{cases} 1 & \text{if } \tau_{t-h}^i \leq h \\ 0 & \text{otherwise;} \end{cases}$$

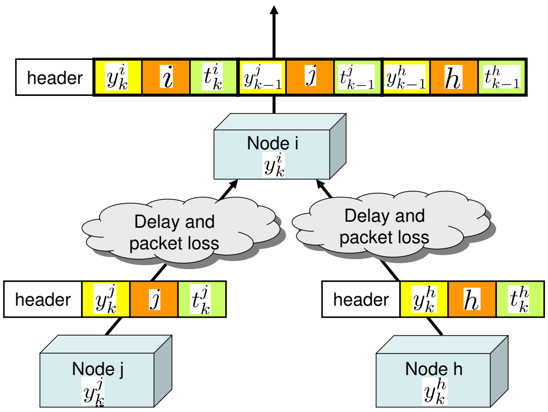

Fig. 2. Data packet structure transmitted by each sensor node for the measurement aggregation strategy.

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.

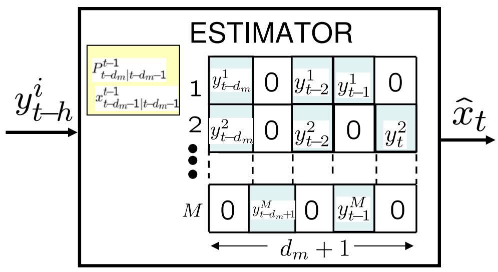

Fig. 3. Memory data structure required at the base station for the measurement aggregation strategy.

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

$$\hat{x}^t_{t-d_m-1|t-d_m-1} = \hat{x}^{t-1}_{t-d_m-1|t-d_m-1},\ P^t_{t-d_m|t-d_m-1} = P^{t-1}_{t-d_m|t-d_m-1} \tag{3}$$

$$\hat{x}^t_{k+1|k} = A\hat{x}^t_{k|k}, \qquad k = t-d_m, \ldots, t-1, t \tag{4}$$
$$P^t_{k+1|k} = AP^t_{k|k}A^T + Q \tag{5}$$
$$\hat{x}^t_{k|k} = \hat{x}^t_{k|k-1} + K_{t,k}(\tilde{y}_{t,k} - C_{t,k}\hat{x}^t_{k|k-1}) \tag{6}$$
$$P^t_{k|k} = P^t_{k|k-1} - P^t_{k|k-1}C^T_{t,k}(C_{t,k}P^t_{k|k-1}C^T_{t,k} + R_{t,k})^{\dagger}C_{t,k}P^t_{k|k-1} \tag{7}$$
$$K_{t|k} = P^t_{k|k-1}C^T_{t,k}(C_{t,k}P^t_{k|k-1}C^T_{t,k} + R_{t,k})^{\dagger} \tag{8}$$

where $()^{\dagger}$ indicates the pseudoinverse operator, $C^T_{t,k} = [\gamma^1_{t,k}C^T_1\ \gamma^2_{t,k}C^T_2\ \ldots\ \gamma^N_{t,k}C^T_N]$, and the matrix $R_{t,k} \in \mathbb{R}^{m\times m}$ is such that the (i,j)-th element is $[R_{t,k}]_{ij} = \gamma^i_{t,k}\gamma^j_{t,k}R_{ij}$. 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 $\tilde{y}_{t,k}$ 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 $O(n^2 + d_m m)$. 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:

$$\hat{x}^t_{k|k} = P^t_{k,k}\left((P^t_{k,k-1})^{-1}\hat{x}^t_{k|k-1} + C^T_{t,k}R^{\dagger}_{t,k}\tilde{y}_{t,k}\right)$$
$$P^t_{k,k} = \left((P^t_{k,k-1})^{-1} + C^T_{t,k}R^{\dagger}_{t,k}C_{t,k}\right)^{-1}$$

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 $R = \mathrm{diag}(R_{11}, R_{22}, \ldots, R_{NN})$. Under this additional assumption the previous equations can be rewritten as

$$\hat{x}^t_{k|k} = P^t_{k,k}\Big((P^t_{k,k-1})^{-1}\hat{x}^t_{k|k-1} + \sum_{i=1}^{N}\gamma^i_{t,k}C_i^T R_{ii}^{-1} y^i_k\Big) \quad (9)$$

$$P^t_{k,k} = \Big((P^t_{k,k-1})^{-1} + \sum_{i=1}^{N}\gamma^i_{t,k}C_i^T R_{ii}^{-1} C_i\Big)^{-1} \quad (10)$$

The terms in the summation of Equation (9) are known as the information vectors and can be computed locally by each node as follows:

$$z^i_k = C_i^T R_{ii}^{-1} y^i_k$$

Note that $z^i_k \in \mathbb{R}^n$ has the same dimension of the state space. Since Equation (9) requires only the sum of all information vectors z^i_k, 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 ξ^i_k 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:

$$\begin{array}{ll} \xi^i_t & = z^i_t \\ \xi^i_k & = \sum_{j\in\mathcal{C}(i)} \nu^j_t \xi^j_k, \quad k = t-d_m,\dots,t-1 \end{array} \quad (11)$$

where ν^j_t is a binary local variable that it 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 $\nu^j_t = 0$, then all the ξ^j_k vectors coming from the j-branch are lost. In other words,

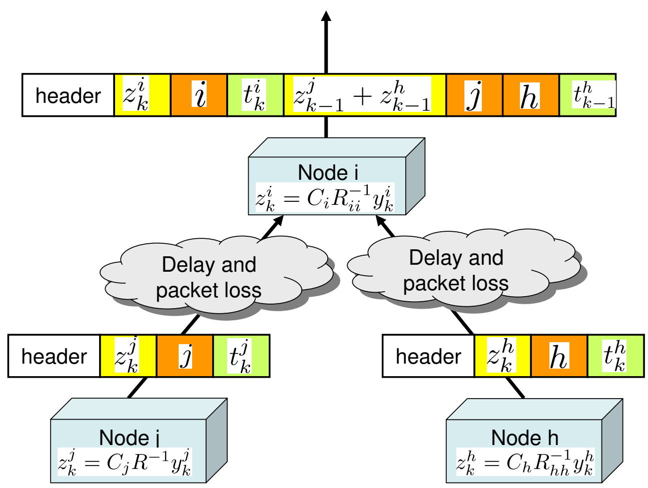

Fig. 4. Data packet structure transmitted by each sensor node for the measurement fusion strategy.

if a packet from a children to a parent is dropped, then the cumulative information vectors from that node are simply not added. Therefore, each sensor node transmit the cumulative information vectors together with the sensor node identities which contributed to it, and the corresponding sampling time, as shown in Figure 4. Using this measurement fusion scheme the cumulative information vectors sent by the root sensor node to the base station can be used to compute the sufficient statistics required by Equation (9). In fact we have:

$$\begin{array}{ll} \zeta_{t,t} & = \xi^1_t = \sum_{i=1}^N \gamma^i_{t,t} C_i^T R_{ii}^{-1} y^i_t \\ \zeta_{t,k} & = \zeta_{t-1,k} + \xi^1_k = \sum_{i=1}^N \gamma^i_{t,k} C_i^T R_{ii}^{-1} y^i_k \end{array} \quad (12)$$

where $k = t-d_m,\dots,t-1$. The identities of the sensor nodes which contributed to $\zeta_{t,k}$ are used to compute $\sum_{i=1}^N \gamma^i_{t,k} C_i^T R_{ii}^{-1} C_i$ necessary for Equation (10). Figure 5 shows the data memory structure required at the base station.

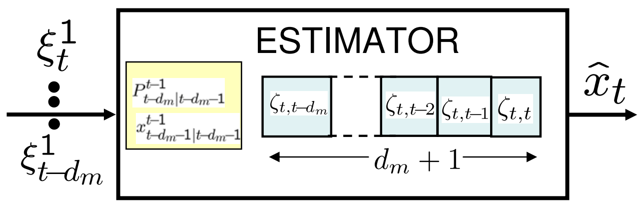

Fig. 5. Memory data structure required at the base station for the measurement fusion strategy.

The state estimate obtained at the base station $\hat{x}^{BS}_{t|t}$ 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 $P^t_{t-d_m|t-d_m}$ 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 $\eta_k := [x_k^T, x_{k-1}^T, .., x_{k-d_m}^T]^T$ the augmented state. Assume also that the communication between the $i-th$ node and the fusion center entails d^i time step delays. To the purpose of estimator design[1], one can think that the fusion center receives the measurements y^i_k of the i-th node with d^i delays. If we denote with $\bar{y}^i_k$ the data available to the fusion center from node i at time k, clearly $\bar{y}^i_k = y^i_{k-d_i}$ holds true. Therefore, an estimator which accounts for the communications delays can be designed based on the augmented model:

$$\begin{array}{rcl} \eta_{k+1} & = & \bar{A}\eta_k + \bar{w}_k \\ \bar{y}^i_k & = & \bar{C}_i \eta_k + \bar{v}^i_k = C_i x_{k-d^i} + v^i_{k-d^i} \quad i = 1,..,n \end{array} \quad (13)$$

where

$$\bar{A} := \begin{bmatrix} A & 0 & \dots & 0 \\ I & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & I & 0 \end{bmatrix}, \quad w_k := \begin{bmatrix} w_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

[1] We shall see that, in fact, measurements are not sent through the network.

and

$$\bar{C}_i := [\underbrace{0,..,0}_{d_i-1}, C_i, \underbrace{0,..,0}_{d_m-d_i-1}].$$

In order to fix notation let us denote with $\hat{\eta}_{k|k}$ the estimator of the augmented state η_k using data up to time k and consider the (steady state) "centralized" Kalman filter:

$$\begin{array}{rcl} \hat{\eta}_{k|k} & = & F\hat{\eta}_{k|k} + \sum_{i=1}^{n} L_i \bar{y}_k^i \\ \bar{y}_k^i & = & \bar{C}_i \eta_k + \bar{v}_k^i \quad i = 1,..,n \end{array} \tag{14}$$

where $L := [L_1,..,L_n] = \bar{A}P\bar{C}^T\left(\bar{C}P\bar{C}^T + R\right)^{-1}$, P is the (augmented) state prediction error covariance which is the solution of the standard Algebraic Riccati Equation [14], and $F = \bar{A} - L\bar{C}$.

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:

$$\begin{array}{rcl} \hat{\eta}_{k|k}^i & = & F\hat{\eta}_{k|k}^i + L_i \bar{y}_k^i \\ \bar{y}_k^i & = & \bar{C}_i \eta_k + \bar{v}_k^i \quad i = 1,..,n \\ \hat{\eta}_{k|k} & = & \sum_{i=1}^{n} \hat{\eta}_{k|k}^i \end{array} \tag{15}$$

where the i-th node is responsible for computing $\hat{\eta}_{k|k}^i$, which we shall call a "partial" estimate of the state η_k. Note that $\hat{\eta}_{k|k}^i$ uses only $\bar{y}_k^i = y_{k-d_i}^i$. Indeed one can think that, at time $k-d_i$, node i computes $\hat{\eta}_{k|k}$ 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.

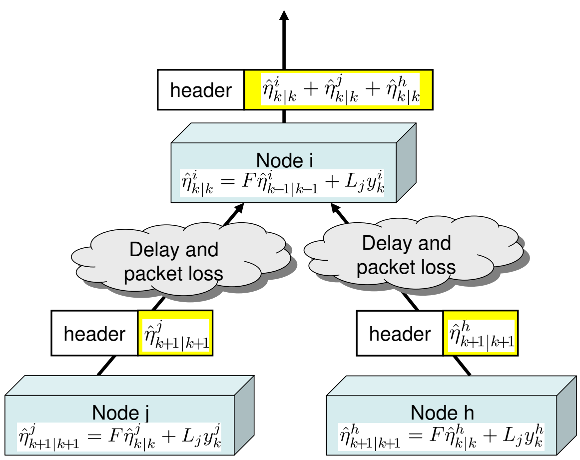

Fig. 6. Data packet structure transmitted by each sensor node for the partial state estimate fusion strategy.

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 $\hat{\eta}_{k|k}^i$. In this scenario, it is not clear how these should be replaced in the computation of the estimator

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

At a first glance letting the $i-$th partial estimator to evolve in open loop, i.e. setting $\eta_{k|k}^i = A\eta_{k-1|k-1}^i$ 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 ℓ-ary trees, i.e. trees where each node has ℓ children, of depth d_m, where the root node has depth zero. We define with the variables b_h, b_a, b_t, b_r 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:

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

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

$$\begin{aligned} N_{ma} &= M_1 b_h + (m b_r + b_t + b_a) M_2 \\ N_{mf} &= M_1 b_h + b_a M_2 + (n b_r + b_t) M_3 \\ N_{sf} &= M_1 (b_h + n b_r) \end{aligned}$$

where N_{ma}, N_{na}, 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: $k = 3, b_h = 4, b_a = 2, b_t = 2, b_d = 6$. In fact, in this example,

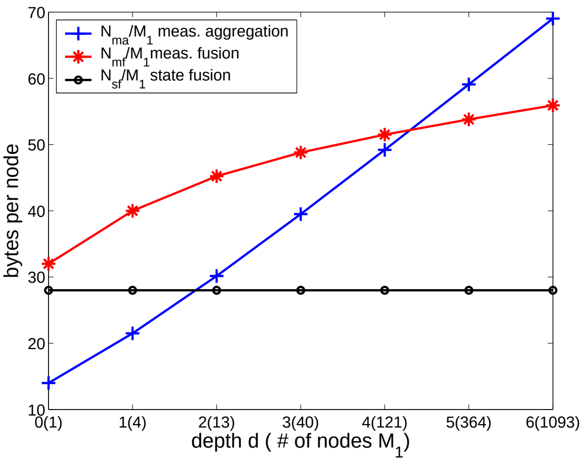

Fig. 7. Average number of bytes transmitted per node per time step as a function of the tree depth d. On the x-axis inside the round parentheses there are also indicated the corresponding total number of sensor nodes in the network.

while the partial state estimate strategy becomes convenient already for a tree composed of only a dozen sensor nodes, the measurement fusion strategy becomes more convenient than the measurement aggregation strategy only when the tree has more than a hundred nodes. These thresholds are a strong function of the ratio n/m, i.e. the larger the ratio the more convenient is to send the raw measurements.

VIII. Conclusions

In this paper we presented three strategies for state estimation in rooted wireless sensor networks subject to packet 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

[1] V. Borkar and P. Varaiya, “Asymptotic agreement in distributed estimation,” *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 650–655, 1982.

[2] J. Tsitsiklis and M. Athans, “Convergence adn asymptotic agreement in distributed decision problems,” *IEEE Transactions on Automatic Control*, vol. 29, no. 1, pp. 42–50, 1984.

[3] D. Castanon and D. Teneketsis, “Further results on the asymptotic agreement problem,” *IEEE Transactions on Automatic Control*, vol. 33, no. 6, pp. 515–523, 1988.

[4] D. Spanos, R. Olfati-Saber, and R. Murray, “Approximate distributed kalman filtering in sensor networks with quantifiable performance,” in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN05*, 2005, pp. 133–139.

[5] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed Kalman filtering based on consensus startegies,” *to appear in IEEE Journal on Selected Areas of Communications*, 2008.

[6] J. Speyer, “Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem,” *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, 1979.

[7] T. Chang, “Comments on “computation and transmission requirements for a decentralized linear-quadratic-gaussian control”,” *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 609–610, 1980.

[8] M. B. A.S. Willsky, D. Castanon, B. Levy, and G. Verghese, “Combining and updating of local estimates and regional maps along sets of one-dimensional tracks,” *IEEE Transactions on Automatic Control*, vol. 27, no. 4, pp. 799–813, 1982.

[9] N. Carlson, “Federated square root filter for decentralized parallel processors,” *IEEE Transactions on Aerospace and Electronics Systems*, vol. 26, no. 3, pp. 517–525, 1990.

[10] J. Wolfe and J. Speyers, “A low-power filtering scheme for distributed sensor networks,” in *In Proc. of the 42th IEEE Conf. on Decision and Contr.*, 2003, pp. 6325–6326.

[11] Y. Bar-Shalom, H. Chen, and M. Mallick, “One-step solution for the general out-of-sequence measurement problem in tracking,” *IEEE Transactions on Aerospace and Electronics Systems*, vol. 40, no. 1, pp. 27–37, 2004.

[12] K.-S. Zhang, X. R. Li, and Y.-M. Zhu, “Optimal update with out-of-sequence observations for distributed filtering,” *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 1992–2004, 2005.

[13] L. Schenato, “Optimal sensor fusion for distributed sensors subject to random delay and packet loss,” in *to appear in Proc. of IEEE Conf. on Decision and Control*, New Orleans, U.S.A., 2007.

[14] B. Anderson and J. Moore, *Optimal Filtering*. Prentice Hall, 1979.

[15] A. Gombani and M. Pavon, “A general Hankel-norm approximation scheme for linear recursive filtering,” *Automatica*, vol. 26, no. 1, pp. 103–111, 1990.