Estimate Aggregation with Delay Constraints in Multihop Wireless Sensor Networks

Haitao Zhang, Huadong Ma
Beijing Key Lab of Intelligent Telecomm. Software and Multimedia,
Beijing University of Posts and Telecomm.,
Beijing, China
Email: zhht.83@gmail.com; mhd@bupt.edu.cn

Xiang-Yang Li
Dept. of Computer Science
Illinois Institute of Technology,
Chicago, USA
Email: xli@cs.iit.edu

***Abstract*—Using wireless sensor networks (WSNs) to observe physical processes for control systems has attracted much attention recently. For some real-time control applications, controllers need to accurately estimate the process state within rigid delay constraints. In this paper, we jointly consider state estimation and scheduling problems in multihop WSNs. For accurately estimating a process state as well as satisfying rigid delay constraints, we propose an in-network estimation approach which includes two coupled parts: the estimation operations performed at sensor nodes and an aggregation scheduling algorithm. Our in-network estimation operation performed at intermediate relay nodes not only optimally fuses the estimates obtained from different sensors but also predicts upper stream sensors' estimates which can not be aggregated to the sink before deadlines. Our estimate-based aggregation scheduling algorithm, which is interference-free, is able to aggregate as much estimate information as possible from a network to a sink within delay constraints. We prove the unbiasedness of in-network estimation, and theoretically analyze the optimality of our approach. Our simulation results corroborate the theoretical results.**

***Keywords*-estimation, aggregation scheduling, delay, wireless sensor networks, cyber-physical systems**

I. Introduction

Because of the advantages of low cost and easy deployment, WSNs are regarded as a more promising means of observing physical world for future Cyber-Physical Systems (CPS), e.g., the central plant heating, ventilation, and air-conditioning (HVAC) system in intelligent building [1] and the tunnel monitoring application [2]. In these control systems, WSNs are responsible for sensing physical processes and gathering the state measurements or estimates of physical processes from physically distributed sensor nodes to a control center or a sink (which we use interchangeably in this paper). On the one hand, the process state can not be obtained directly due to the presence of the process noise and the measurement noise [3]. Therefore, the system must estimate the process state as accurately as possible based on the sensor measurements, and use the estimates as the input of the controller. On the other hand, from the viewpoint of networked control systems (NCSs), significant transmission delay is equivalent to data loss, which leads to performance degradation or even loss of stability of systems. Thus, a key challenge in these systems is to design efficient and effective transmission protocols to satisfy the rigid delay requirement of the control applications under wireless interference constraints.

Currently, in the field of WSNs, the researches of state estimation and protocol design have remained largely separate. In the aspect of state estimation, estimation based on single sensor information over lossy networks has been extensively studied recently [4]−[8]. But these estimation techniques only view the network as a single end-to-end communication channel characterizing some data loss model and adapt to underlying communication protocols passively. In addition, multi-sensor information fusion is usually adopted in WSNs to deal with sensing uncertainty, and many distributed estimation algorithms have been proposed [9]−[12]. However, the current distributed estimation methods are restricted to either single-hop networks or multihop networks without a center, and are not suitable for the multihop networks with one fixed sink. In the aspect of protocol design in WSNs, aggregation scheduling has attracted much attention recently. In WSNs, compared with transmitting raw data, in-network data aggregation is effective in improving delay performance [13] as well as reducing energy consumption [14]. Several protocols have been proposed in the literature for delay-efficient data aggregation in WSN, e.g., [15]−[19]. All these work focus on minimizing the total time of aggregating the whole sensor data from the network to the sink under various interference models. However, these scheduling schemes are not applied to some large-scale networked real-time systems in which aggregating all the sensor information can not satisfy the rigid delay constraints. In [20] and [21], the authors jointly addressed estimation and communication problems in CPS, and designed an adaptive-reliability transport protocol. But they did not consider the real-time issue of control systems.

In this paper, we consider a state estimation problem with stringent delay constraints in large-scale WSNs, and this is an important issue for real-time networked control applications. The object is to obtain an optimal state estimate at the sink within hard delay constraints through the collaboration of a set of distributed sensors and the sink. Compared with prior work, we jointly design the estimation algorithm and the scheduling protocol in multihop WSNs,

978-0-7695-4361-1/11 $26.00 © 2011 IEEE
DOI 10.1109/ICCPS.2011.9

and propose a novel in-network estimation approach for accurately estimating a state as well as satisfying delay constraints. Our in-network estimation is a progressive estimate fusion technique in which every intermediate node on the route calculates an optimal fusion estimate based on the received information from its children nodes and its own measurements and then uses the fused estimate as a scheduling unit. Unlike the minimum delay aggregation scheduling, our aggregation scheduling is to gather maximum sensor information at the sink within stringent delay constraints, and thus collecting information from more sensors implies the reduced estimation error.

The main contributions of the paper are as follows. First, we give a formulation of optimal in-network estimation with delay constraints in multihop WSNs, which addresses the co-design of the estimate fusion operation and the aggregation scheduling. Second, we present an in-network estimate fusion method which not only optimally fuses the estimates obtained from different sensors but also predicts upper stream sensors' estimates which can not be aggregated to the sink before deadlines. Third, under the protocol interference model, we propose an aggregation scheduling algorithm for optimally estimating a process state and satisfying delay constraints, and we theoretically prove that the proposed scheduling algorithm is interference-free. Fourth, we prove the unbiasedness of our in-network estimation, and analyze the optimality of our approach. In addition to theoretical analysis, extensive simulations are conducted, and the results corroborate our theoretical analysis.

The rest of the paper is organized as follows. Section II outlines the related work. Section III formulates the problem. The in-network estimation operations are presented in Section IV. We propose an estimate aggregation scheduling algorithm in Section V. Section VI analyzes the performance of our in-network estimation approach. Section VII presents the simulation results. We conclude the paper in Section VIII.

II. Related Work

A. State Estimation in WSNs

Estimation over lossy networks has been well studied in recent years. Sinopoli *et al.* [4] considered the problem of performing Kalman filtering with intermittent observations whose arrival is modeled as a random process. Smith *et al.* [5] proposed a suboptimal but computationally efficient estimator that can be applied when the arrival process is modeled as a Markov chain. In [6] and [7], the authors proposed to estimate the process state (or encode the sensor measurements) at the sensor side of the link without assuming any statistical model for the data loss process. Schenato *et al.* [8] designed the optimal estimators over lossy networks under the TCP-like and UDP-like communication protocols respectively. All the methods mentioned above treat a network of communication links as a single end-to-end link with some data loss model. By using the memory and processing ability of intermediate nodes, Gupta *et al.* [22] proposed a recursive algorithm for information processing at the nodes of the network so that the estimator can calculate the optimal state estimate for any packet-dropping process, but the strategy is only for the single source case.

Distributed estimation is an important signal processing problem for wireless sensor networks. If the sensors in networks exchange and fuse their sensing information, the resulting estimate can be better than that based on the sensor own measurements. Roumeliotis *et al.* [9] decomposed a single Kalman filter into a number of smaller communicating filters for the multirobot localization problem. Sun *et al.* [10] proposed a multi-sensor optimal information fusion criterion weighted by matrices in the linear minimum variance sense. Based on this optimal fusion criterion, a general multi-sensor optimal information fusion decentralized Kalman filter with a two-layer fusion structure was given for discrete time linear stochastic control systems with multiple sensors and correlated noises. However, all of the above referenced works are restricted to single-hop networks. For many applications, large-scale sensor networks are needed to collect data from a wide area. Distributed estimation for multihop networks has also attracted strong interests recently. Based on consensus averaging, Schizas *et al.* [11] presented a distributed Kalman smoother state estimator. Speranzon *et al.* [12] proposed a new distributed algorithm for cooperative estimation of a slowly time-varying signal using WSNs. However, the above mentioned algorithms for multihop networks are all iterative in nature, and they are not suitable for the network with one fixed sink because their convergence can not be guaranteed within any given time window.

B. Delay-Efficient Scheduling for Data Aggregation

Data aggregation is considered to be an effective method for improving delay performance in multihop wireless networks. In multihop WSNs, every intermediate node combines all received data with its own data according to an aggregation function, and transmits the aggregated data rather than the raw data in networks. Consequently, the data aggregation time from the network to a distinguished sink and the energy consumption are reduced because the data needed to be scheduled in networks is reduced [13] [14].

Minimum delay data aggregation in WSNs under various interference models has been proven to be NP-hard [15], and several approximation algorithms have been proposed recently, e.g., [15]$-$[19]. Chen *et al.* [15] proposed an algorithm to generate a collision-free schedule with a latency bound of $(\Delta-1)R$, where Δ is the maximum node degree and R is the network radius. Huang *et al.* [16] proposed a centralized aggregation scheduling algorithm with the latency bound $23R+\Delta-18$, and the algorithm is based on a simple primary interference model: no node can send and receive simultaneously. Under the protocol interference model, Wan *et al.* [18] proposed three centralized data aggregation methods for networks when nodes have the same transmission radius and interference radius. An efficient distributed algorithm that produces an interference-free schedule for data aggregation was proposed in [17], and the delay is at most $24D + 6\Delta + 16$ time-slots where D is the network diameter. Xu *et al.* [19] proposed a distributed aggregation scheduling method generating interference-free schedules with an upper-bound on delay of $16R + \Delta - 14$ time-slots where R is the radius of the network..

There have been lots of work on delay-efficient aggregation scheduling in WSNs and the object is to minimize the total time of aggregating the whole sensor data from the network to the sink, but there is no work on estimate aggregation scheduling for large-scale network systems in which the object is to gather maximum sensor information at the sink within stringent delay constraints.

III. Models and Problem Formulation

A. Network Model

Consider a multihop WSN $G = (V, E)$ where V is the set of n nodes in the network and E is the set of communication links. Assume a node cannot send and receive data simultaneously. To let two links transmit simultaneously, we must ensure they are interference-free. In the protocol interference model [23] on which our work is based, we assume that each node has an interference range r_I. A receiver v of a link uv is interfered by the signal from another sender p if $\| p - v \| \leq r_I$.

We assume that there is an aggregation tree Q rooted at a sink node $v_n \in V$. There are many ways to construct an aggregation tree. One example is the distributed approach presented by Wan et al. [24]. But this aggregation tree may not be optimal for the purpose of in-network estimation. Finding the best tree for state estimation remains an open research topic.

B. Data Model

In this paper, we consider a common discrete-time data model which characterizes the linear process state and observation and can be motivated by many practical applications. We assume sufficient bits per data packet so that the quantization error is negligible. This assumption makes sense if the communication packet provides enough bits for transmitting data so that the effect of quantization error is dominated by the effect of the process and the measurement noises [22].

The discrete-time linear dynamical process considered in this paper is modeled by [4] [8]

$$x(k+q) = Ax(k) + w(k) \tag{1}$$

where $x(k) \in \mathbf{R}^p$ is the process state vector at time k, A is the state-transition matrix of the process, q is the sampling period, and $w(k) \in \mathbf{R}^p$ is the zero-mean white Gaussian process noise with covariance matrix $R_w > 0$ and uncorrelated across time. The initial state $x(0)$ is assumed to be independent of $w(k)$ and to have mean zero and covariance matrix $R(0)$. The process state to be estimated has different physical interpretation in different systems, e.g., the temperature vector whose component is the local temperature at different locations, and the position vector of a target to be tracked [3]. The state-transition matrix A characterizes the temporal correlation between the states of two consecutive sampling time slots. The observations about the common state are collected by physically distributed sensors according to the measurement model

$$y_i(k) = B_i x(k) + v_i(k) \tag{2}$$

where $y_i(k) \in \mathbf{R}^{p_i}$ is the measurement output vector generated by the sensor i at time k, B_i is the observation matrix of the sensor i, and $v_i \in \mathbf{R}^{p_i}$ is the measurement noise of the sensor i which is assumed to be white, zero-mean, Gaussian with covariance matrix $R_i > 0$ $(1 \leq i \leq n-1)$ and is uncorrelated across time and sensors and independent of the process noise $w(k)$.

C. Problem Formulation

For the accurate estimation of the state $x(k)$, the sink needs to gather the sensing information as much as possible. Moreover, the sensing information collected at time k should be received by the sink before a given deadline. Therefore, we address the problem from two coupled aspects. The first aspect is an effective aggregation scheduling which can gather much sensing information as possible from the network to the sink within every scheduling period (which will be defined later). The other aspect is the estimation operation which is performed by each node and responsible for processing the sensor information aggregated from the upper stream sensors.

In our in-network estimation approach, we transmit the estimates of the state instead of the measurements. In the network G, when the sensor i $(1 \leq i \leq n-1)$ samples a new measurement $y_i(k)$ at time k, it computes a state estimate $\hat{x}_i(k)$ based on its local measurements and the estimates received from its upper stream sensors, and then forwards the estimate $\hat{x}_i(k)$ to the next hop node along the aggregation tree. The estimate error covariance matrix of the node i is expressed as

$$P_i(k) = \mathrm{E}\left\{ [x(k) - \hat{x}_i(k)] \, [x(k) - \hat{x}_i(k)]^{\mathrm{T}} \right\} \tag{3}$$

where E is the expectation, and T denotes the transpose. Finally, based on the acquired estimates from the network, the sink performs an unbiased fusion estimation.

Let $A, B \subset V$ and $A \cap B = \emptyset$. We say that data are aggregated from A to B in one time-slot if all the nodes in A transmit data simultaneously in one time-slot and all the data are received by some nodes in B without interference, and A is called a sender set. In this paper, the delay constraint is set to the sampling period q. This is reasonable because the sensor information sensed at time k is outdated when there is the new sensing information obtained right after time $k+q$. For simplicity, we assume that q is an integer multiple of one time-slot assigned to each scheduled sensor, the networked system starts running at time 0, and the clocks of all nodes are synchronized. Within every sampling period q, since there is no new sensing information, the object of the scheduling is unchanged and q can also be termed as a scheduling period.

Then, in the *l*-th round of schedule ($l = 0, 1, \ldots$), an aggregation schedule for in-network estimation can be defined as a sequence of sender sets $S_1, S_2, \ldots, S_q$ satisfying the following conditions:

1) $S_i \cap S_j = \emptyset$, $\forall i \neq j$ and $i, j \in \{1, 2, \ldots, q\}$;
2) The estimates are aggregated from S_k to $V - \bigcup_{i=1}^{k} S_i$ at time-slot k, for all $k = 1, 2, \ldots, q$, and the estimates of the state of time lq are aggregated to the sink v_n before time $(l+1)q$.

Notice that we discard the condition $\bigcup_{i=1}^{p} S_i = V - v_0$ required in the other delay-efficient aggregation scheduling problems [17] [19]. This is because we can not gather all the estimates from the network to the sink before the deadlines when q is so small relative to the network size. The condition 1 is to ensure that a node participates in the data aggregation at most once in one scheduling period.

Given the multihop WSN G, the in-network estimation problem with delay constraints is to jointly design the estimation operation $\hat{x}_i(k)$ $(1 \leq i \leq n)$ of each node and an aggregation schedule $S_1, S_2, \ldots, S_q$ such that the estimate $\hat{x}_n(t)$ at the sink v_n satisfies the following goals:

1) Unbiasedness, $\mathrm{E}[\hat{x}_n(k)] = \mathrm{E}[x(k)]$;
2) Optimality, minimizing the trace of fusion estimate error covariance, $\min\{\mathrm{tr}[P_n(k)]\}$, where $\mathrm{tr}[\cdot]$ denotes the trace of matrix.

IV. In-network Estimation

In this section, we present the estimation operation $\hat{x}_i(t)$ $(1 \leq i \leq n)$ of every node in the network. Before giving the details, we outline the overall process of our in-network estimation approach in every scheduling period as follows:

1) Every sensor node samples the dynamical process state at time lq and performs a local estimation based on its own measurements obtained before time lq ($l = 0, 2, \ldots$). Then, it waits for being scheduled.
2) If a leaf node is scheduled, it transmits the local estimate to its parent directly. If a relay node is scheduled, it first performs an optimal information fusion based on the estimates received from its child nodes and its own local estimate, and then transmits the fused estimate to the next-hop node.
3) At the deadline of every scheduling period, the sink calculates an optimal estimate based on the previously received information.

Note that we consider a large-scale multihop network scenario in which not all the estimates of the state of time lq can be aggregated to the sink before time $(l+1)q$. Therefore, we need to design an optimal estimate fusion method which is capable of dealing with the two cases: the complete upper stream information and the incomplete upper stream information. In addition, the selection of the estimate fusion scheme depends on the scheduling, and this will be discussed in Section V.

A. Estimation at the Leaf

The local estimator at the leaf node adopts the standard Kalman filter. We respectively define $\hat{x}_i(k) \triangleq \mathrm{E}[x(k)|Y_i(k)]$ and $P_i(k) \triangleq \mathrm{E}[(x(k)-\hat{x}_i(k))(x(k)-\hat{x}_i(k))^{\mathrm{T}}|Y_i(k)]$ as the local filtering estimate and the estimate error covariance of the node i at time k where $Y_i(k) \triangleq \{y_i(0), \ldots, y_i(k)\}$ ($k = lq$). The prediction step of the filter estimation is given by

$$\hat{x}_i(k|k-q) \triangleq A\hat{x}_i(k-q) \tag{4}$$

$$P_i(k|k-q) \triangleq AP_i(k-q)A^T + R_w. \tag{5}$$

The correction step is given by

$$K_i(k) \triangleq P_i(k|k-q)B_i^T \left(B_i P_i(k|k-q) B_i^T + R_i\right)^{-1} \tag{6}$$

$$\hat{x}_i(k) = \hat{x}_i(k|k-q) + K_i(k)\left[y_i(k) - B_i\hat{x}_i(k|k-q)\right] \tag{7}$$

$$P_i(k) = P_i(k|k-q) - K_i(k)B_iP_i(k|k-q). \tag{8}$$

After the local filter estimation, the leaf node i uses the message $msg_i(k) = (\hat{x}_i(k), P_i(k), k)$ as the scheduling unit of the *l*-th scheduling period, and waits for being scheduled.

B. Estimation at the Relay and the sink

As a relay sensor node i, once it obtains a new measurement at sampling time k, it first performs the Kalman filtering based on its own measurements according to the equations (4)-(8), and the local estimate $\hat{x}_i^t(k)$ and the local error covariance $P_i^t(k)$ are acquired. We define CS_i as the children set of the node i and define $C_i = CS_i \cup \{i\}$. During the *l*-th scheduling period, the relay i may receive the messages $msg_j(k)$ ($j \in CS_i$) from its children before being scheduled. Now we will give an estimate fusion method which combines the previous received estimation information and its local estimation results. The fusion method is based on the optimal fusion algorithm weighted by matrices in the linear minimum variance sense. Based on our variable definitions, we represent the following optimal fusion theorem introduced in [25] and [26].

Theorem 1: Let $\hat{x}_j^t(k)$ ($j \in C_i$) be unbiased estimates of a *p*-dimensional stochastic vector $x(k)$. Let the estimate errors be $\tilde{x}_j^t(k) = x(k) - \hat{x}_j^t(k)$. Assume that $\tilde{x}_{j_1}^t(k)$ and $\tilde{x}_{j_2}^t(k)$ ($j_1 \neq j_2$) are uncorrelated, and the estimate error variance matrix is denoted by $P_j^t(k)$. Then, the optimal fusion estimator with matrix weights is given by

$$\hat{x}_i(k) = \sum_{j \in C_i} w_j(k) \hat{x}_j^t(k) \tag{9}$$

where the optimal matrix weights $w_j(k)$ are computed by

$$w_j(k) = \left[\sum_{m \in C_i} \left(P_m^t(k) \right)^{-1} \right]^{-1} \left(P_j^t(k) \right)^{-1}, \tag{10}$$

and the corresponding minimal fusion error variance matrix is

$$P_i(k) = \left[\sum_{m \in C_i} \left(P_m^t(k) \right)^{-1} \right]^{-1}. \tag{11}$$

Theorem 1 gives an optimal fusion criterion in the linear minimum variance sense, and we use the equations (9)-(11) as the estimate fusion method executed at relays. However, at some relays, not all the children state estimates of time k can be aggregated to them at the l-th scheduling period due to the rigid delay constraint. For example, if the relay i does not receive the message from the child j before i is scheduled at the l-th scheduling period, i can predict the child j's estimate and the covariance of time k based on the previously obtained information from the child j. Then, at the relay i, the fused estimate and the corresponding covariance of the child j are respectively computed by

$$\hat{x}_j^t(k) = (1 - \gamma_j(k)) \hat{x}_j^t(k|k-q) + \gamma_j(k) \hat{x}_j(k), \tag{12}$$

and

$$P_j^t(k) = (1 - \gamma_j(k)) P_j^t(k|k-q) + \gamma_j(k) P_j(k), \tag{13}$$

where $\hat{x}_j^t(k|k-q)$ and $P_j^t(k|k-q)$ are respectively the predicted estimate and the corresponding predicted covariance of time k based on those of the previous scheduling period and they are both initialized to 0, and $\gamma_j(k) \in \{0, 1\}$ is a binary parameter. In the prediction process, we respectively use the prediction equations (4) and (5) for computing $\hat{x}_j^t(k|k-q)$ and $P_j^t(k|k-q)$. In the l-th scheduling period, if i can receive the $msg_j(k)$ before being scheduled, $\gamma_j(k)$ is set to 1, else $\gamma_j(k)$ is set to 0. $\gamma_j(k)$ is jointly determined by the scheduling, the network topology and q, and it will be further discussed in Section V.

At the sink, we also adopt the above estimate fusion method expressed by (9)-(13) except that we use CS_n instead of C_n in the fusion equations (9)-(11).

Remark 1: The state estimates obtained from different sensors are not conditionally independent in general due to the common process noise. Therefore, the proposed estimate fusion algorithm is suboptimal. However, it is more computation-efficient than the complicated fusion method that takes the correlation of the state estimates into account, and is a more attractive in-network computation scheme for resource-constraint WSNs.

V. Aggregation Scheduling with Delay Constraints for Estimation

In this section, we design an effective interference-free estimate aggregation scheduling algorithm EASDC for satisfying delay constraints and accurately estimating the state. Our estimate aggregation scheduling algorithm is based on an aggregation tree which can either be a BFS (breadth-first-search) tree or that constructed by existing methods, like [24] [17]. For each node i in the aggregation tree Q, let p_i be i's parent, let NS_i be the set of i's one-hop neighbors except p_i. In our algorithm, every node i should maintain the following local variables.

1) Number of Children: NoC_i, the number of i's children nodes in the aggregation tree Q.
2) Children Number of i's Parent: $CNoP_i$, the number of the children nodes of i's parent in the aggregation tree Q.
3) Time-Slot to First Transmit: $TSFT_i$, the assigned time-slot at which i send its data to its parent for the first time.
4) Node Scheduling Period: NSP_i, the scheduling period of i such that i is scheduled once every NSP_i time-slots after time $TSFT_i$.
5) Children Set: CS_i, the node set of i's children such that the set elements are arranged according to the descending order of the size of the subtree rooted at $j \in CS_i$.
6) Indicator Array of Available Time-Slot: $IAATS_i[\cdot]$, a binary array such that if a child node of i can transmit data without interference in time-slot t, $IAATS_i[t]$ is 1, else $IAATS_i[t]$ is 0. $IAATS_i.size$ is the size of the array.
7) $PSN_i = \{p_j\}_{j \in NS_i - CS_i}$.
8) $NSC_i[j] = NS_j$ ($j \in CS_i$).
9) $RANK_i = (level, i)$ where $level$ is the hop distance of i to the root. The ranks of nodes are compared using lexicographic order.

For accurately estimating the state, the estimate information should be gathered at the sink as much as possible within delay constraints. We define $PSNuNSC_i = PSN_i \bigcup (\bigcup_{j \in CS_i} NSC_i[j])$. The scheduling time of a node i is determined by $TSFT_i$ and NSP_i. To determine the schedules for all the nodes in a sensor network, we use a up-bottom time-slot assignment method: assign the time-slot to nodes level by level starting from the root level. The node i assigns the time-slot to its children according to the known interference conditions expressed by $IAATS_i$, and sends the message SCHDL($tsft, nsp$) to its children and then send the message SCHDL-CMPLT($tsft, nsp$) to PSN_i and the corresponding $NSC_i[\cdot]$. Upon receiving a message

SCHDL($tsft, nsp$), a child j sets $tsft$ and nsp to $TSFT_j$ and NSP_j, respectively. Upon receiving a message SCHDL-CMPLT($tsft, nsp$), a node updates its $IAATS$ according to it. Once a node finish the schedule assignment task for its children, it sends the message FINISHED-SCHDL to its $PSNuNSC$. Upon receiving a message FINISHED-SCHDL from v, a node delete v from its $PSNuNSC$, and if its $RANK$ is smaller than that of every node in its $PSNuNSC$, it begins to assign time-slots to its children. The details of our scheduling algorithm EASDC are shown in Algorithm 1.

Algorithm 1 Estimate Aggregation Scheduling with Delay Constraints EASDC.

Input: A network G, the aggregation tree Q rooted at the node n, and the delay constraint q;
Output: $TSFT_i$ and NSP_i for every node i ($1 \leq i < n$)
Every node i initializes NoC_i, $CNoP_i$ ($CNoP_n \leftarrow 1$), $RANK_i$, CS_i, PSN_i, and $NSC_i[\cdot]$ based on the tree Q; and $TSFT_i \leftarrow 0$; $NSP_i \leftarrow q$; allocates space for the binary array $IAATS_i[\cdot]$ whose size is $q \cdot \max\{NoC_i, CNoP_i\}$ and each element of the array is initialized to 1, $DONE_i \leftarrow FALSE$;
Root n calls TAAC($CS_n, NoC_n, IAATS_n, q+1, NSP_n, PSN_n, NSC_n$);
for Each node i, upon receiving a message SCHDL($tsft, nsp$) **do**
  $TSFT_i \leftarrow tsft$; $NSP_i \leftarrow nsp$; $DONE_i \leftarrow TRUE$;
  $m \leftarrow 0$;
  while $tsft + m \cdot nsp \leq IAATS_i.size$ **do**
    $IAATS_i[tsft + m \cdot nsp] \leftarrow 0$; $m \leftarrow m + 1$;
  if There is available time-slot based on $IAATS_i$, $tsft + nsp$ and nsp **then**
    if $RANK_i < RANK_j$ for each $j \in PSNuNSC_i$ **then**
    Call TAAC($CS_i, NoC_i, IAATS_i, TSFT_i, NSP_i, PSN_i, NSC_i$).
for Each node i, upon receiving a message SCHDL-CMPLT($tsft, nsp$) **do**
  $m \leftarrow 0$;
  while $tsft + m \cdot nsp \leq IAATS_i.size$ **do**
    $IAATS_i[tsft + m \cdot nsp] \leftarrow 0$; $m \leftarrow m + 1$;
for Each node i, upon receiving a message FINISHED-SCHDL from node k **do**
  if $k \in PSNuNSC_i$ **then**
    $PSNuNSC_i \leftarrow PSNuNSC_i - \{k\}$;
  if $DONE_i = TRUE$ **then**
    if There is available time-slot based on $IAATS_i$, $tsft + nsp$ and nsp **then**
      if $RANK_i < RANK_j$ for each $j \in PSNuNSC_i$ **then**
        Call TAAC($CS_i, NoC_i, IAATS_i, TSFT_i, NSP_i, PSN_i, NSC_i$).

TAAC (shown in Algorithm 2) is the time allocation procedure for children. The time-slot assignment principle of our algorithm is as follows. Consider a node i with the assigned $TSFT_i$ and NSP_i, we assume $NACS$ is the set of i's children which have not been assigned time-slots and noc is the size of $NACS$. If a child $j \in NACS$ has the maximum size of subtree rooted at j compared to the other children in $NACS$, it may aggregate more estimate information before time $TSFT_i$. Therefore, we should set the maximum available time-slot t before $TSFT_i$ to $TSFT_j$, and NSP_j is equal to NSP_i. Based on $IAATS$, $TSFT$ and NSP, the available time-slot t is a time-slot such that $t < TSFT$ and $IAATS[t + m \cdot NSP] = 1$ where $m \in \{m' | 0 \leq m' \leq (IAATS.size - TSFT)/NSP, m' \in \mathbb{Z}\}$. If there is only one available time-slot t before time $TSFT_i$ and $noc > 1$, we can not schedule all the children in $NACS$ within one node scheduling period NSP_i. Then, in every node scheduling period, we choose one child j in $NACS$ in turn to be scheduled and NSP_j is set to $NSP_i \cdot noc$. If there is no one available time-slot and $noc > 0$, node i can not gather any information from its children in the current scheduling period. However, by exploiting the temporal correlation of the process state, we can predict the estimates based on the previously obtained estimate information from $NACS$, and the predicted estimate information can contribute to the estimation accuracy. Therefore, if the estimate information of $NACS$ can be scheduled before time $TSFT_i + NSP_i$, it is also useful for estimation accuracy.

Algorithm 2 Time Allocation Algorithm for Children TAAC($CS, noc, IAATS, tsft, nsp, PSN, NSC$).

for Select a node j from CS according to the descending order of the size of the subtree rooted at j.
  if There is no available time-slot based on $IAATS$, $tsft$ and nsp, and $noc > 0$ **then**
    $tsft \leftarrow tsft + nsp$;
  if There is only one available time-slot t based on $IAATS$, $tsft$ and nsp, and $noc > 1$ **then**
    Send the message SCHDL($t, nsp \cdot noc$) to j;
    Send the message SCHDL-CMPLT($t, nsp \cdot noc$) to
    $PSN \bigcup NSC[j]$;
    $CS \leftarrow CS - \{j\}$;
    $i \leftarrow 1$;
    for Select a node k from CS **do**
      Send the message SCHDL($t + nsp \cdot i, nsp \cdot noc$) to k;
      Send the message SCHDL-CMPLT($t + nsp \cdot i, nsp \cdot noc$) to $PSN \bigcup NSC[k]$;
      $i \leftarrow i + 1$;
      $CS \leftarrow CS - \{k\}$.
  else
    Find the maximum available time-slot t based on $IAATS$, $tsft$ and nsp;
    $m \leftarrow 0$;
    while $t + m \cdot nsp \leq IAATS.size$ **do**
      $IAATS[t + m \cdot nsp] \leftarrow 0$; $m \leftarrow m + 1$;
    Send the message SCHDL(t, nsp) to j;
    Send the message SCHDL-CMPLT(t, nsp) to
    $PSN \bigcup NSC[j]$;
    $CS \leftarrow CS - \{j\}$;
    $noc \leftarrow noc - 1$.
Send the message FINISHED-SCHDL to $PSNuNSC$.

Figure 1 illustrates the scheduling results of our algorithm for a small sensor network. There is a label $x(y, z)$ beside

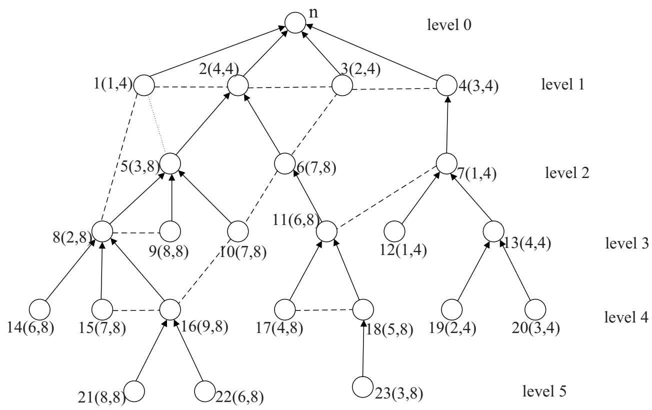

Figure 1. An example of the result of our scheduling algorithm.

a node where x is the node's ID, y and z are respectively the $TSFT$ and NSP of the node. The solid lines represent edges in the aggregation tree and the dotted lines represent the other edges in the graph. Node n is the sink. The delay constraint q is set to 4. At first, the nodes at the level 1 are assigned the scheduling time. Because the subtree of node 2 has a maximal size than the subtrees of node 1, 3 and 4 (the size of the subtree of a leaf node is 0), node n sends the message SCHDL$(4, 4)$ to node 2, and sends the message SCHDL-CMPLT$(4, 4)$ to nodes 1 and 3. Then, node n respectively sends the messages SCHDL$(3, 4)$, SCHDL$(2, 4)$ and SCHDL$(1, 4)$ to nodes 4, 3 and 1 in sequence, and respectively sends the corresponding SCHDL-CMPLT messages to its PSN and the corresponding NSCs. Upon receiving the message SCHDL$(4, 4)$, node 2 respectively sets 4 and 4 to its $TSFT$ and NSP, and waits for the FINISHED-SCHDL message from 1. Upon receiving the messages, node 2 calls TAAC to assign scheduling time to its children. Since there is only one available time-slot 3 and two nodes 5 and 6 need to be assigned time, node 2 respectively sends the messages SCHDL$(3, 4 \times 2)$ and SCHDL$(3 + 4 \times 1, 4 \times 2)$ to nodes 5 and 6. In this way, all the nodes are assigned scheduling time.

Remark 2: The sink estimates the state every q time-slots. However, when the size of the network is so large that not all the estimates can be aggregated from the network to the sink within q time-slots. In the above example (shown in Fig. 1), the estimate of node 12 can not be aggregated to the sink before the deadline. Node 12 still sends its estimate to its parent node 7 because node 7 can predict the estimate of node 12 in the next scheduling period based on the estimate sent in the current scheduling period. Therefore, the parameter $\gamma_{12}(lq)$ of node 7 is 0 ($l = 0, 1, \ldots$). Similarly, the parameters $\gamma_5(2lq)$ and $\gamma_6((2l+1)q)$ of node 2 are 1, and $\gamma_5((2l+1)q)$ and $\gamma_6(2lq)$ are 0. In summary, the parameters $\gamma_x(t)$ are determined by schedule, network topology and delay constraints, and thus the in-network estimation is not stochastic.

Theorem 2: Algorithm EASDC generates an interference-free aggregation schedule.

Proof: We prove that the resulting schedule is interference-free by contradiction. For each node i in the aggregation tree, we define $CoN_i = \bigcup_{j \in NS_i - CS_i} CS_j$. Suppose there is an interference in time-slot k_0, then there must exist two nodes v_1 and v_2 both have k_0 as their schedules, and v_1's parent or v_2's parent hears two messages in k_0. There are four cases in this situation. The first case is $p_1 = p_2$. In EASDC, the schedules of a node i is assigned by p_i according to $IAATS_i[\cdot]$ by calling TAAC. If p_1 (or p_2) sets v_1's schedule to k_0, it can not set v_2's schedule to k_0. Here comes the contradiction. The second case is $v_2 \in CoN_1$, and this means $p_1 \in PSN_{p_2}$ and $p_2 \in NSC_{p_1}[1]$. If $RANK_{p_1} \leq RANK_{p_2}$, p_2 can not assign time-slots to v_2 until it receives a FINISHED-SCHDL message from p_1, else p_1 receives a FINISHED-SCHDL message from v_2 before assigning time-slots to v_1. Therefore, v_1 and v_2 can not obtain the same schedule anyway. The three case, which is $v_1 \in CoN(v_2)$, can be proved by the same way as the second case. The fourth case is $v_2 \in \bigcup_{u \in CS_1} CS_u$ (or $v_1 \in \bigcup_{u \in CS_2} CS_u$). This means $p_2 \in NSC_{p_1}[1]$ (or $p_1 \in NSC_{p_2}[2]$), and p_2 (or p_1) can receive the SCHDL-CMPLT message from p_1 (or p_2) before setting schedule to v_2 (or v_1). Therefore, if the schedule k_0 is set to v_1 (or v_2), v_2 (or v_1) can not obtain the schedule k_0. Till now we have completed the proof of Theorem 2. ■

VI. Performance Analysis

A. Estimation Unbiasedness

Theorem 3: By using the in-network estimation approach given in Section IV and Section V, the estimate obtained by the sink is unbiased for every scheduling period, namely $\mathrm{E}[\hat{x}_n(k)] = \mathrm{E}[x(k)]$.

Proof: First, we consider the relay nodes whose children are just the leaves in the aggregation tree Q, and we define these relays as 1-relay nodes. If a 1-relay node i can receive all the estimates $\hat{x}_j(k)$ of its children within the *l*-th scheduling period by using the algorithm EASDC ($k = lq$), the estimates to be fused at node i are all the Kalman filtering estimates of time k, and thus the fused estimate $\hat{x}_i(k)$ obtained by using the equations (9) and (10) is unbiased according to Theorem 1. If some children can not send their estimates to node i within the *l*-th scheduling period, their estimates can be predicted based on previously received estimates by using the equation (4), and the predicted estimates are unbiased obviously. Then, the fused estimate based on these partially predicted estimates is still unbiased according to Theorem 1. Therefore, 1-relay nodes perform an unbiased estimation.

Next, we consider the relay nodes whose children are the leaves or 1-relay nodes, and these relays are defined as 2-relay nodes. The estimates of 2-relay's children are unbiased.

Therefore, no matter whether a 2-relay node can receive all the estimates from its children within one scheduling period, the 2-relay node performs an unbiased estimation according to the above unbiasedness proof about 1-relay nodes.

Third, we consider the relay nodes whose children are the leaves, 1-relay nodes or 2-relay nodes, and these relays are defined as 3-relay nodes. Similarly, we can conclude the estimate of a 3-relay node for every scheduling period is unbiased. Obviously, the sink is a 3-relay node, and the theorem has been proved till now. ■

B. Optimality Analysis

Now we analyze the optimality of our in-network estimation approach.

Definition 1: Given an tree consisting of a node set, the root centered set (RCS) is a connected node subset which contains the root of the tree.

Lemma 1: Suppose that V_1 is a RCS of the aggregation tree Q. Let $P_i^t(k)$ be the estimate error covariance matrix of the node v_i's filtering estimation for the *l*-th scheduling period ($v_i \in V_1$ and $k = lq$). If all the estimation information of the nodes in V_1 can be aggregated to the sink v_n within one scheduling period by using our in-network estimation approach presented in Section IV and Section V, we have $P_n(k) = [\sum_{i \in V_1} (P_i^t(k))^{-1}]^{-1}$.

The proof of lemma 1 is omitted due to space limitation.

Theorem 4: Suppose that the node subsets V_1 and V_2 are two RCSs of the aggregation tree Q and satisfy $V_1 \subset V_2$. By using our in-network estimation approach presented in Section IV and Section V, $P_n^{V_1}(k)$ and $P_n^{V_2}(k)$ are the estimate error covariance matrices of the sink v_n for the *l*-th scheduling period based on the estimates obtained from V_1 and V_2, respectively ($k = lq$). Then, we have $\mathrm{tr}[P_n^{V_1}(k)] \geq \mathrm{tr}[P_n^{V_2}(k)]$.

Proof: Since $V_1 \subset V_2$, there exists a node v_1 that satisfy $v_1 \in V_2 - V_1$. We define $V_1^1 \triangleq V_1 \bigcup \{v_1\}$. Let $P_i^t(k)$ (> 0) be the error covariance of the v_i's filtering estimate for the *l*-th scheduling period. By using our estimate aggregation approach, if the estimation information can be aggregated to v_n within one scheduling period, we have $P_n^{V_1}(k) = [\sum_{v_i \in V_1} (P_i^t(k))^{-1}]^{-1} > [\sum_{v_i \in V_1^1} (P_i^t(k))^{-1}]^{-1} = P_n^{V_1^1}(k)$ according to Lemma 1, else there exists a v_1's ancestor node v_1^a whose estimation information can be aggregated to v_n within one scheduling period and v_1^a may obtain a predicted estimation information $P_1^p(k)$ (≥ 0) of v_1 based on the previously received information, and thus we also have $P_n^{V_1}(k) \geq P_n^{V_1^1}(k)$ according to Lemma 1. If $V_1^1 = V_2$, we have $\mathrm{tr}[P_n^{V_1}(k)] \geq \mathrm{tr}[P_n^{V_2}(k)]$, else there exists a node v_2 that satisfy $v_2 \in V_2 - V_1^1$, and then we define $V_1^2 \triangleq V_1^1 \bigcup \{v_2\}$. Based on the above set construction method, we can obtain a set sequence $V_1^1 \subset V_1^2 \subset \cdots \subset V_1^r$ ($r = |V_2 - V_1|$ and $V_1^r = V_2$). According to the above proof, we have $P_n^{V_1}(k) \geq P_n^{V_1^1}(k) \geq \cdots \geq P_n^{V_1^r}(k) = P_n^{V_2}(k)$. Therefore, we have $\mathrm{tr}[P_n^{V_1}(k)] \geq \mathrm{tr}[P_n^{V_2}(k)]$. ■

Theorem 4 shows that estimation accuracy of the sink can be improved by aggregating the more estimation information of the nodes from the network. In addition, the new estimation information is more useful for improving the estimation accuracy than the old one obviously. Therefore, we can evaluate optimality of our in-network estimation approach through measuring how many nodes can send their updated estimation information to the sink within delay constraints. Next, we will give the overall upper-bound on the number of the nodes that can send their updated estimation information to the sink within delay constraints. Here the overall upper-bound refers to maximum number of the nodes whose estimates for the state of time k can be aggregated to the sink by any method before time $k + q$.

Theorem 5: Suppose that n_i is number of nodes at the *i*-th level of the aggregation tree Q. Under any interference model, the overall upper-bound of the number of the nodes whose estimates for the state of time k can be aggregated to the sink by any method before time $k + q$ is $\sum_{i=1}^{q} n_i$.

Proof: The upper-bound $\sum_{i=1}^{q} n_i$ immediately follows from the fact that no matter what algorithm is implemented and no matter what interference model we will use, the sink can gather at most all the estimates of the nodes whose level is less than or equal to q within the delay constraint q. ■

Theorem 6: Under the protocol interference model, there is a placement of nodes such that the number of nodes whose estimates for the state of time k are aggregated to the sink before time $k + q$ by using our estimate aggregation can achieve the upper-bound $\sum_{i=1}^{q} n_i$ provided in Theorem 5.

Theorem 6 can be proved by using the construction method, and the detail is omitted due to space limitation.

Theorem 7: Under the protocol interference model, the number of the nodes whose estimates for the state of time k are aggregated to the sink before time $k+q$ by using our estimate aggregation is at least $\min\{q, n\}$.

Theorem 7 is obvious according to our scheduling algorithm EASDE.

VII. Simulation Results

In our simulation, we randomly deploy sensors into a region of $200m \times 200m$. The number of nodes is fixed to 500. All sensors have the same transmission radius which is fixed to 25m. An aggregation tree rooted at the sink is constructed by using the BFS method. In fact, other aggregation tree construction methods, like the method proposed in [17], are also suitable. We consider the discrete-time linear dynamical system (1) and (2) with $A = \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix}$ and $R_w = I_2$. The measurement matrix B_i is chosen from the following matrices:
$B^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $B^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $B^3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$.
Assume the maximum level number of the aggregation is H. If the level number of node v_i is less than $\frac{1}{3} \cdot H$, B_i is set to B^1, else if the level number of node v_i is more than

$\frac{2}{3} \cdot H$, B_i is set to B^3, else B_i is set to B^2. The covariance matrix R_i of the measurement noise is chosen from $10 \cdot I_2$, $20 \cdot I_2$ and $30 \cdot I_2$ randomly. The initial state x_0 and error covariance matrix $R(0)$ are respectively set to $[10 \quad 1\ ]^{\mathrm{T}}$ and $10 \cdot I_2$.

For the performance comparison with our in-network estimation approach, we implement a non-aggregation estimation approach named NAE. In NAE, each sensor node first performs a filter estimation based on its own measurements, and then sends the estimates to the sink along the aggregation tree without data fusion at relays. For satisfying the delay constraint, the estimates are gathered by the sink level by level starting from the lowest level.

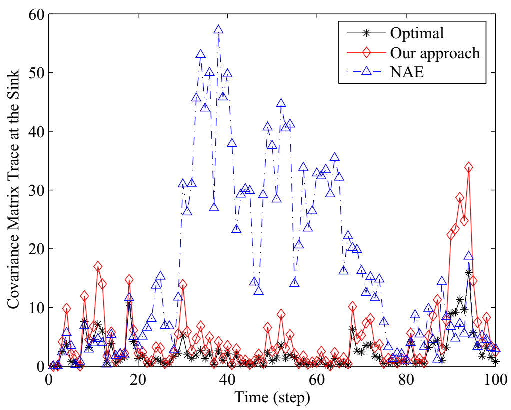

Figure 2. Covariance matrix trace at the sink for three methods.

In Fig. 2, the delay constraint q is fixed to 100. The figure compares the traces of the estimate error covariance matrix of the sink with every estimation time step using the three methods. Here the optimal method is the fusion-based estimation method that performs the optimal estimate fusion given in Theorem 1 on the estimates of the whole network. On the one hand, it can be seen that our in-network estimation approach has a similar estimation performance to the optimal method. This is because that though the sink can not gather all the real-time estimates from the whole network due the delay constraint, our in-network estimation approach has the ability to compensate for the estimates loss of the remote sensors by exploiting the temporal correlation of the state. On the other hand, our in-network estimation approach can gather more estimate information than NAE within the delay constraint that the estimation performance of our method is better than that of NAE in the most of the time. More importantly, the stability of the estimation accuracy of our approach outperforms that of NAE observably, and this performance criteria is critical to control applications.

Fig. 3 and Fig. 4 show the traces of the estimate error covariance matrix of the sink with every estimation time step for our in-network estimation approach and NAE, respectively. From Fig. 3, if the delay constraint is reduced, the estimation accuracy of our approach will degrade, and the degradation rate is slow. However, as illustrated in Fig. 4, we can see the estimation accuracy of NAE is influenced by the delay constraints greatly. Once the delay constraint is reduced, the estimation accuracy will degrade dramatically. Therefore, our in-network estimation approach can achieve a better tradeoff between estimation accuracy and delay constraints.

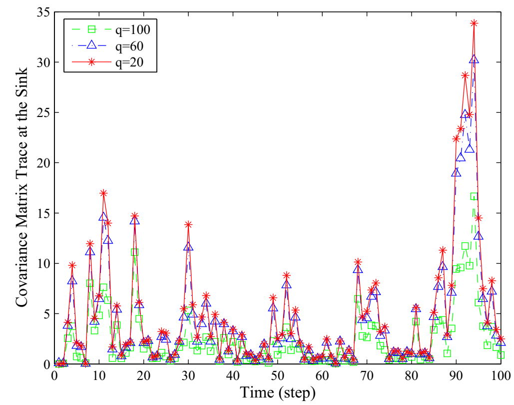

Figure 3. Covariance matrix trace at the sink for our in-network estimation approach.

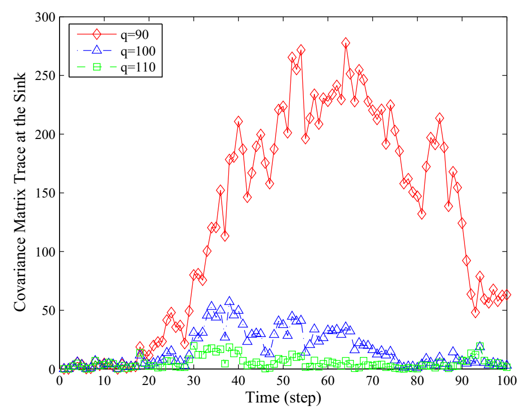

Figure 4. Covariance matrix trace at the sink for NAE.

VIII. Conclusions

For WSNs-based control applications, a novel in-network estimation approach was proposed for state estimation with delay constraints in multihop WSNs. For accurately estimating a process state as well as satisfying rigid delay constraints, we addressed the problem through jointly designing in-network estimation operations and an aggregation scheduling algorithm. Our in-network estimation operation performed at relays not only optimally fuses the estimates obtained from different sensors but also predicts the upper stream sensors' estimates which can not be aggregated to the sink before deadlines. Our estimate aggregation scheduling algorithm, which is interference-free, is able to aggregate as much estimate information as possible from a network to a sink within delay constraints. The unbiasedness of our in-network estimation has been proved, and the theoretical analysis about the estimation optimality and the simulation results show that our approach can achieve a considerably high estimation accuracy.

ACKNOWLEDGMENT

The research reported in this paper was supported by the National Basic Research Program of China (973 Program) under grant No. 2011CB302701; The National Natural Science Foundation of China under Grant No. 60833009; China National Funds for Distinguished Young Scientists under Grant No. 60925010; NSF CNS-0832120; NSF CNS-1035894; The program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

REFERENCES

[1] A. Wheeler, "Commercial Applications of Wireless Sensor Networks Using ZigBee," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 70-77, Apr. 2007.

[2] L. Mottola and G. P. Picco, "Programming Wireless Sensor Networks with Logical Neighborhoods: A Road Tunnel Use Case," *ACM Sensys*, pp. 393-394, Nov. 2007.

[3] A. Ribeiro, I. Schizas, S. Roumeliotis, and G. Giannakis, "Kalman Filtering in Wireless Sensor Networks," *IEEE Control Systems Magazine*, vol. 30, no. 2, pp. 66-86, Apr. 2010.

[4] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman Filtering With Intermittent Observations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453-1464, Sep. 2004.

[5] S. Smith and P. Seiler, "Estimation with lossy measuraments: jump estimators for jump systems," *IEEE Trans. on Automatic Control*, vol. 48, no. 12, pp. 1453-1464, Dec. 2003.

[6] Y. Xu and J. Hespanha, "Estimation under controlled and uncontrolled communications in networked control systems," *44th IEEE CDC-ECC*, pp. 842-847, Dec. 2005.

[7] V. Gupta, D. Spanos, B. Hassibi, and R. M Murray, "Optimal LQG control across packet-dropping links," *Systems and Control Letters*, vol. 56, no. 6, pp. 439-446, June 2007.

[8] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of Control and Estimation over Lossy Networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163-187, Jan. 2007.

[9] S. I. Roumeliotis and G. A. Bekey, "Distributed Multirobot Localization," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 781-795, Oct. 2002.

[10] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017-1023, June 2004.

[11] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy linksłPart II: Distributed estimation and smoothing of random signals," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1650-1666, Apr. 2008.

[12] A. Speranzon, C. Fischione, and K. H. Johansson, "Distributed and collaborative estimation over wireless sensor networks," *45th IEEE CDC*, pp. 1025-1030, Dec. 2006.

[13] R. Tan, G. Xing, B. Liu, and J. Wang, "Impact of Data Fusion on Real-Time Detection in Sensor Networks," *30th IEEE Real-Time Systems Symposium*, pp. 323-332, Dec. 2009.

[14] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, "Tina: A scheme for temporal coherency-aware in-network aggregation," *ACM MobiDE*, pp. 69-76, Sep. 2003.

[15] X. Chen, X. Hu, and J. Zhu, "Minimum data aggregation time problem in wireless sensor networks," *1st Intl Conference on Mobile Ad-hoc and Sensor Network*, pp. 133-142, May 2005.

[16] S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks," *INFOCOM 2007*, pp. 366-372, May 2007.

[17] B. Yu, J. Li, and Y. Li, "Distributed Data Aggregation Scheduling in Wireless Sensor Networks," *INFOCOM 2009*, Apr. 2009.

[18] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, X. Jia, "Minimum-Latency Aggregation Scheduling in Multihop Wireless Networks," *ACM MobiHoc 2009*, pp. 185-193, May 2009.

[19] X. Xu, S. Wang, X. Mao, S. Tang, and X. Li, "A Delay Efficient Algorithm for Data Aggregation in Multi-hop Wireless Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, 2010.

[20] H. Ahmadi and T. Abdelzaher, "An Adaptive-Reliability Cyber-physical Transport Protocol for Spatiotemporal Data," *RTSS 2009*, pp. 238-247, Dec. 2009.

[21] H. Ahmadi, T. Abdelzaher, and I. Gupta, "Congestion Control for Spatio-temporal Data in Cyber-physical Systems," *ICCPS 2010*, pp. 89–98, Apr. 2010.

[22] V. Gupta, A. Dana, J. Hespanha, R. Murray, and B. Hassibi, "Data Transmission Over Networks for Estimation and Control," *IEEE Trans. on Automatic Control*, vol. 54, no. 8, pp. 1807-1819, Aug. 2009.

[23] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388-404, Mar. 2000.

[24] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *INFOCOM 2002*, pp. 1597-1603, June 2002.

[25] N. A. Carlson, "Federated Square Root Filter for Decentralized Parallel Processes," *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 517-525, May 1990.

[26] S. Sun, "Multi-Sensor Optimal Information Fusion Kalman Filter for Discrete Multichannel ARMA Signals," *IEEE International Symposium on Intelligent Control*, pp. 377-382, Oct. 2003.