In-Network Estimation with Delay Constraints in Wireless Sensor Networks

Haitao Zhang, Huadong Ma, *Member*, *IEEE*, Xiang-Yang Li, *Senior Member*, *IEEE*, and Shaojie Tang, *Student Member*, *IEEE*

Abstract—The use of wireless sensor networks (WSNs) for closing the loops between the cyberspace and the physical processes is more attractive and promising for future control systems. For some real-time control applications, controllers need to accurately estimate the process state within rigid delay constraints. In this paper, we propose a novel in-network estimation approach for state estimation with delay constraints in multihop WSNs. For accurately estimating a process state as well as satisfying rigid delay constraints, we address 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 the different sensors but also predicts the upper stream sensors' estimates which cannot 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 the network to the sink within delay constraints. We proved the unbiasedness of in-network estimation, and theoretically analyzed the optimality of our approach. Our simulation results corroborate our theoretical results and show that our in-network estimation approach can obtain significant estimation accuracy gain under different network settings.

Index Terms—Wireless sensor networks, estimation, aggregation scheduling, delay constraint, cyber-physical systems, networked control systems

1 Introduction

Cyber-physical Systems (CPSs) are a promising new class of systems characterizing seamless, fully synergistic integration of cyber capabilities and physical processes, and can find application in a broad range of areas [1], [2]. Unlike traditional cyber systems which are usually the passive information receptors in the physical world, CPSs make cyber resources interact with the physical world automatically and intelligently. Because of the advantages of low cost and easy deployment, WSNs are regarded as a more promising means of observing physical world for CPSs [3], e.g., smart building applications [4], [5], [6], waste water processing systems [7], and disaster monitoring systems [8], [9].

In these WSN-based CPSs whose architecture is depicted in Fig. 1, a set of sensor nodes distributed in a surveillance area perform the state sampling of dynamical processes, and the spatiotemporal sampling data are collected from the sensor nodes to one or more control centers (or sinks which we use interchangeably in this paper) by using a multihop wireless network. Then, based on the collected sensor information, controllers make appropriate control decisions which drive actuators to change the physical processes. In these networked systems, on the one hand, before making correct control design, the controller residing in the control center needs to know the process state to be controlled as accurately as possible. However, the process state cannot be obtained directly due to the presence of the process noise and the measurement noise [10]. Therefore, the system must estimate the process state based on the sensor measurements, and use the estimates as the input of the controller. On the other hand, the presence of the wireless network in the feedback control loop means that the data packets can be randomly dropped and delayed. From the viewpoint of networked control systems (NCSs), the significant transmission delay is equivalent to the data loss, which leads to the performance degradation or even loss of the stability of the systems. Thus, the sensor information must be gathered at the control center within very stringent time constraints. A key challenge in these systems is then 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 WSN-based CPSs, the researches of the state estimation and the protocol design have remained largely separate. In the aspect of the state estimation, a method for compensating for the data loss caused by communication is to improve the estimators so that they can estimate the dynamic process states with intermittent observations as accurately as possible. The estimation problem based on single sensor information over lossy networks has been extensively studied recently [11], [12], [13], [14], [15]. But, these estimation techniques only

- *H. Zhang and H. Ma are with the Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Mailbox 139, Haidian District, Beijing 100876, China.*
  E-mail: zhht.83@gmail.com, mhd@bupt.edu.cn.
- *X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, Sturt Building, 229C, 10 West 31st Street, Chicago, IL 60616, and the Dalian University of Technology, Dalian, China.*
  E-mail: xli@cs.iit.edu.
- *S. Tang is with the Department of Computer Science, Illinois Institute of Technology, Sturt Building, 229C, 10 West 31st Street, Chicago, IL 60616.*
  E-mail: stang7@iit.edu.

Manuscript received 30 Dec. 2011; revised 22 Mar. 2012; accepted 24 Mar. 2012; published online 10 Apr. 2012.
Recommended for acceptance by K. Wu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-12-0949.
Digital Object Identifier no. 10.1109/TPDS.2012.122.

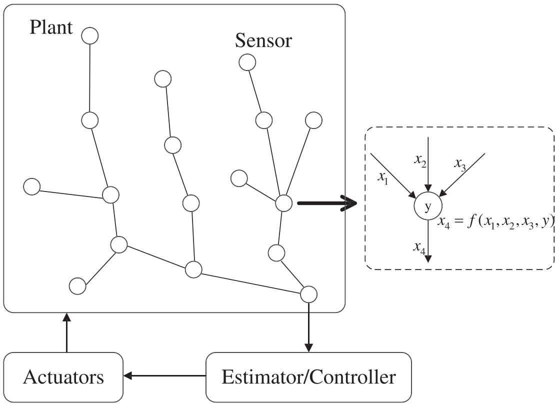


Fig. 1. Architecture of WSN-based CPS.

view the network as a single end-to-end communication channel characterizing some data loss model and adapt to the underlying communication protocols passively. In addition, multisensor information fusion is usually adopted in WSNs to deal with the sensing uncertainty, and many distributed estimation algorithms have been proposed [16], [17], [18], [19], [20]. However, the current distributed estimation methods are restricted to either single-hop networks or multihop networks without a center, and are thus not suitable for the multihop networks with one fixed sink. In the aspect of the protocol design in WSNs, aggregation scheduling has attracted much attention recently. In WSNs, compared with transmitting raw data, in-network data aggregation (whose operation is depicted in Fig. 1) is effective in improving delay performance as well as reducing energy consumption [21], [22], [23]. Several protocols have been proposed in the literature for delay-efficient data aggregation in WSNs, e.g., [24], [25], [27], [26], [28]. All these work focus on minimizing the total delay of aggregating the whole sensor data from the network to the sink under various interference models. However, these scheduling schemes cannot be applied to some large-scale networked real-time systems in which aggregating all the sensor information cannot satisfy the rigid delay constraints.

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. Our objective is to obtain an optimal state estimate at the sink within rigid delay constraints through the collaboration of a set of distributed sensors and the sink. Compared with prior work, we jointly design an estimation method and a scheduling algorithm in multihop WSNs, and propose a novel in-network estimation approach for accurately estimating the 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 information received from its child nodes in the aggregation tree and its own measurements. Unlike the minimum delay aggregation scheduling, the objective of our aggregation scheduling is to gather maximum sensor information at the sink within stringent delay constraints, and this is because that we can reduce the estimation error through collecting the more information from the sensor networks.

The main contributions of our work are as follows: First, to the best of our knowledge, this is the first work to use codesign of the estimate fusion operation and the aggregation scheduling to address the estimate aggregation problem with delay constraint in WSNs. Second, we propose an in-network estimate fusion method to optimally fuse the different estimates. Our in-network estimation operation cannot only optimally fuse the estimates received from different sensors but also predict the upper stream sensors' estimates which cannot be aggregated to the sink before deadlines. Third, under the protocol interference model, we design 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 its optimality.

The rest of the paper is organized as follows: Section 2 formulates an in-network estimation problem. The distributed in-network estimation operations are presented in Section 3. We propose an estimate aggregation scheduling algorithm in Section 4. Section 5 analyzes the performance of our in-network estimation approach. Section 6 presents the simulation results. Section 7 reviews the related work. We conclude the paper in Section 8.

2 Problem Formulation

2.1 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 [29] on which our work is based, 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 $\parallel p - v \parallel \leq r_I$.

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. [30]. 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.

2.2 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 [31].

The discrete-time linear dynamical process considered in this paper is modeled by Schenato et al. [11], [15]

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

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 the 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 [10]. 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 the physically distributed sensors according to the measurement model

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

where $y_i(k) \in \mathbf{R}^{p_i}$ is the measurement output vector generated by the sensor v_i at time k, B_i is the observation matrix of the sensor v_i, and $w_i \in \mathbf{R}^{p_i}$ is the measurement noise of the sensor v_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 process noise $w(k)$.

2.3 Problem Formulation

For accurately estimating 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 is 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 $v_i (1 \leq i \leq n-1)$ 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 v_i is expressed as

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

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, we divide the time of the each round of scheduling period into q time slots. In one scheduling period, S_i is defined as the sensor node set all the member nodes of which synchronously send packets to their corresponding parent nodes in the ith time slot $(i \in \{1, 2, \ldots, q\})$. The work of scheduling is to assign the time slot to each node in the network. In other words, the scheduling problem is how to determine the node set S for each time slot during which all the member nodes of S synchronously send packets to their corresponding parent nodes.

Thus, in the lth round of schedule $(l = 0, 1, \ldots)$, an aggregation schedule for in-network estimation can be defined as a sequence of sender node 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 the 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 [27], [28]. This is because we cannot gather all the estimates from the network to the sink before the deadline when q is so small relative to the network size. 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 $\hat{x}_n(k)$ 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.

3 Distributed 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 distributed in-network estimation approach in every scheduling period. A flowchart version of the in-network estimation approach is provided in Fig. 2. As shown in the figure, in an aggregation tree, every node takes the following actions:

1. The leaf node samples the dynamical process state at time $l \cdot q$ and performs a local estimation based on its own measurements obtained before time $l \cdot q (l = 0, 1, 2, \ldots)$. Then, it waits for being scheduled. If a leaf node is scheduled, it transmits the local estimate to its parent directly.
2. For the relay node, first, it also samples the dynamical process state at time $l \cdot q$ and performs a local estimation based on its own measurements obtained before time $l \cdot q (l = 0, 1, 2, \ldots)$. Then, it performs an optimal information fusion based on the estimates

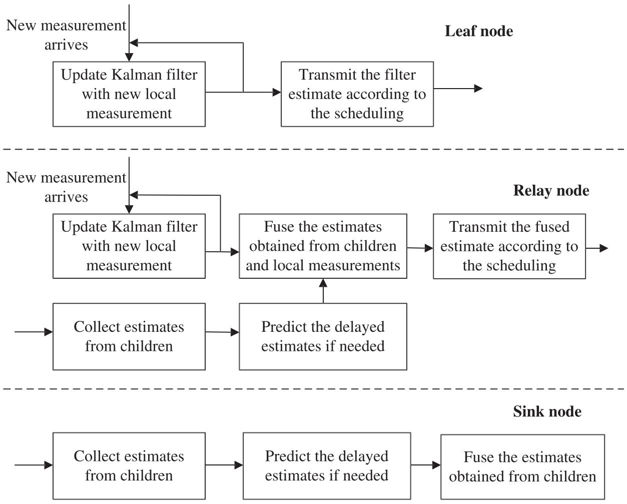

New measurement arrives
Leaf node
Update Kalman filter with new local measurement
Transmit the filter estimate according to the scheduling
New measurement arrives
Relay node
Update Kalman filter with new local measurement
Fuse the estimates obtained from children and local measurements
Transmit the fused estimate according to the scheduling
Collect estimates from children
Predict the delayed estimates if needed
Sink node
Collect estimates from children
Predict the delayed estimates if needed
Fuse the estimates obtained from children

Fig. 2. Summary of in-network estimation.

received from its child nodes and its own local estimate. If the estimates cannot be received from its child nodes in the current scheduling period, the relay node predicts the estimates based on the previously received estimates from its child nodes, and use the predicted estimates to calculate the fused one. Third, when a relay node is scheduled, it transmits the fused estimate to the next-hop node.

3. The sink node continuously collects the estimates from its child nodes. At the deadline of every scheduling period, the sink predicts the estimates based on the previously received estimates from its child nodes, and then calculates an optimal estimate based on the previously received information and the predicted estimates.

Note that we consider a large-scale multihop network scenario in which only part of the state estimates of time $l \cdot q$ can be aggregated to the sink before time $(l+1) \cdot q$. 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 4. The details of the estimation operation performed at each sensor node are present in the following sections.

3.1 Estimation at the Leaf

The local estimator at the leaf node adopts the standard Kalman filter. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements [32]. As such, the equations for the Kalman filter fall into two groups: predictor equations and corrector equations. The predictor equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The corrector equations are responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

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 v_i at time k where $Y_i(k) \triangleq \{y_i(0),\ldots,y_i(k)\}$ $(k = l \cdot q)$. The predictor equations of the filter estimation, which project the state and covariance estimates forward from time step $k-q$ to step k, are 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 corrector equations are given by

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

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

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

The first task during the correction step is to compute the Kalman gain, $K_i(k)$. The next step is to actually measure the process to obtain $y_i(k)$, and then to generate an a posteriori state estimate by incorporating the measurement as in (7). The final step is to obtain an a posteriori error covariance estimate via (8). After each prediction and correction step pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates.

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

3.2 Estimation at the Relay and the Sink

For the relay sensor node v_i, once it obtains a new measurement at sampling time k, it first performs the Kalman filtering based on its own measurements according to (4)-(8), and the local estimate $\hat{x}_i^t(k)$ and the local error covariance $P_i^t(k)$ are acquired consequently. We define CS_i as the children set of the node v_i and define $C_i = CS_i \cup \{i\}$. During the lth scheduling period, the relay v_i may receive 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 previously 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 [33] and [34].

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

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

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

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

and

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

where $\hat{x}^t_j(k|k-q)$ and $P^t_j(k|k-q)$ are, respectively, the predicted estimate and the corresponding predicted covariance of time k 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 (4) and (5) for computing $\hat{x}^t_j(k|k-q)$ and $P^t_j(k|k-q)$. In the lth scheduling period, if node i can receive $msg_j(k)$ 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 4.

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 (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.

4 Decentralized Aggregation Scheduling with Delay Constraints

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 breadth-first-search (BFS) tree or that constructed by the existing methods proposed in [30] and [27]. In algorithm EASDC, we adopt a top-down time-slot assignment mode. In the aggregation tree, each node locally determines the schedules of its child nodes based on the previously received messages. The proposed scheduling algorithm performed in each node does not need any global network information, and thus it is decentralized.

For 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 should maintain the following local variables:

1. Number of children: NoC_i, the number of i's child nodes in the aggregation tree Q.
2. Children number of i's parent: $CNoP_i$, the number of the child 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 by the sink as much as possible within delay constraints. We define

$$PSNuNSC_i = PSN_i \bigcup \left(\bigcup_{j\in CS_i} NSC_i[j]\right).$$

The scheduling time of the node i is determined by $TSFT_i$ and NSP_i. To determine the schedules for all the nodes in the sensor network, we use a top-down 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 message SCHDL($tsft, nsp$) to its children and then sends message SCHDL-CMPLT($tsft, nsp$) to PSN_i and the corresponding $NSC_i[\cdot]$. Upon receiving message SCHDL($tsft, nsp$), the child j sets $tsft$ and nsp to $TSFT_j$ and NSP_j, respectively. Upon receiving message SCHDL-CMPLT($tsft, nsp$), a node updates its $IAATS$ according to the message. Once a node finish the schedule assignment task for its children, it sends message FINISHED-SCHDL to its $PSNuNSC$. Upon receiving message FINISHED-SCHDL from v, a node deletes 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 $i$ ($1 \le i < n$)
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 $Q$;
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;
$TSFT_i \leftarrow 0$; $NSP_i \leftarrow q$; $DONE_i \leftarrow FALSE$;
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 $tsft + m \cdot nsp \le IAATS_i.size$ 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 $tsft + m \cdot nsp \le IAATS_i.size$ 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 the node i with the assigned $TSFT_i$ and NSP_i, we assume $NACS$ is the set of i's children to which the time slot has not been assigned and noc is the size of $NACS$. If the 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 $m \in \{m' | 0 \le m' \le (IAATS.size - TSFT)/NSP, m' \in \mathbb{Z}\}$. If there is only one available time slot t before time $TSFT_i$ and $noc > 1$, we cannot schedule all the children in $NACS$ within one node scheduling period NSP_i. Then, in every 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 $noc > 0$, the node i cannot 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 $t + m \cdot nsp \le IAATS.size$ 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 $PSNuNSC$.
```

In our aggregation scheduling algorithm, every node allocates the transmission schedule to its children by calling Algorithm 2 TAAC. Algorithm 2 TAAC assigns the time slots to the nodes based on the previously received messages and the local topology information. The time slot assignment operations begin to be performed at the root node. The root node also calls TAAC to determine the schedule of its children. So, the proposed scheduling algorithm performed in each node does not need any global network information.

Fig. 3 illustrates the scheduling results of our algorithm for a small sensor network. There is a label $x(y, z)$ beside a node where x is the node's ID, y and z are, respectively, $TSFT$ and NSP of the node. The solid lines represent the 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

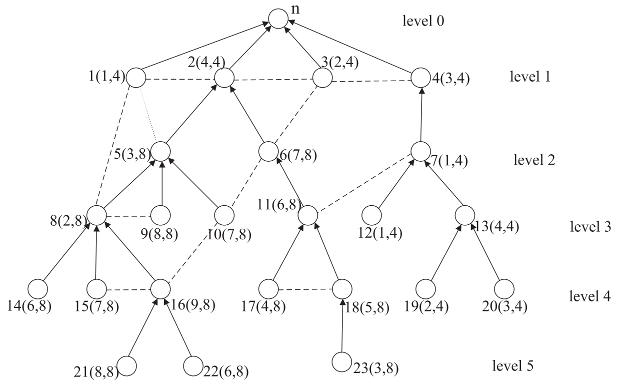


Fig. 3. An example of the result of our scheduling algorithm.

size of the subtree of a leaf node is 0), node n sends message SCHDL(4,4) to node 2, and sends 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 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 the 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 message SCHDL$(3, 4 \times 2)$ and message SCHDL$(3 + 4 \times 1, 4 \times 2)$ to nodes 5 and 6. In this way, all the nodes are assigned the 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. 3), the estimate of node 12 cannot 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 ($l = 0, 1, \dots$). 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 the schedule, the network topology and the 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 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 cannot 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 cannot 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 cannot 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) cannot obtain the schedule k_0. Till now, we have completed the proof of Theorem 2. □

Remark 3. In this work, we assume the aggregation tree is given, and the local topology information for each node is known because it can be obtained in the routing construction process. In the schedule construction process, we adopt a top-down time-slot assignment scheme. The scheduling algorithm begins from the root node, and then is performed by the nodes level by level in the aggregation tree. Each node determines the schedule of its children based on the previously received messages and its local topology information. The information only needs to propagate from the upper level to the lower level without information feedback. In the aggregation tree, since all the nodes make schedules for its child nodes one by one in the worst case, the time complexity of our aggregation scheduling algorithm EASDC is $O(n)$ where n is the number of nodes in the network.

5 Performance Analysis

5.1 Estimation Unbiasedness

Theorem 3. *By using the in-network estimation approach given in Sections 3 and 4, 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 1-relay node i can receive all the estimates $\hat{x}_j(k)$ of its children within the lth scheduling period by using the algorithm EASDC ($k = l \cdot q$), 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 (9) and (10) is unbiased according to Theorem 1. If some children cannot send their estimates to node i within the lth scheduling period, their estimates can be predicted based on previously received estimates by using (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, one-relay nodes perform an unbiased estimation.

Next, we consider the relay nodes whose children are the leaves or one-relay nodes, and these relays are defined as two-relay nodes. The estimates of two-relay's children are unbiased. Therefore, no matter whether a two-relay node can receive all the estimates from its children within one scheduling period, the two-relay node performs an unbiased estimation according to the above unbiasedness proof about one-relay nodes.

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

5.2 Optimality Analysis

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

Definition 1. *Given a 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 lth scheduling period ($v_i \in V_1$ and $k = l \cdot q$). 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 Sections 3 and 4, we have $P_n(k) = [\sum_{i\in V_1}(P_i^t(k))^{-1}]^{-1}$.*

Proof. Because V_1 is a RCS of the aggregation tree Q, it is also a tree obviously. Given relay node j in V_1, we assume T_j is a V_1's subtree rooted at node j. In the following proof, we use the one-relay, two-relay, and three-relay definitions which are given in the proof of Theorem 3. Because the sink does not sample the process state, we assume $P_n^t(k) = \infty$. We will prove that whether relay node j is a one-relay, two-relay, or three-relay, we always have $P_j(k) = [\sum_{i\in T_j}(P_i^t(k))^{-1}]^{-1}$.

First, we consider the node j is a one-relay node. Since the children of node j are just the leaves and all the estimates of j's descendants can be aggregated in node j within one scheduling period, we have $P_j(k) = [\sum_{i\in T_j}(P_i^t(k))^{-1}]^{-1}$ according to Theorem 1.

Second, we consider the node j is a two-relay node. We assume $V_{j,c}^1$ is the j's children set in which the elements are all the one-relay nodes and the j's children set $V_{j,c}^{leaf}$ consists of the leaf nodes. Similarly, according to Theorem 1 and the above proof, we have

$$\begin{aligned} &P_j(k)\\ &= \left[\left(P_j^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^{leaf}} \left(P_l^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^1}\left(P_l^t(k)\right)^{-1}\right]^{-1}\\ &= \left[\left(P_j^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^{leaf}} \left(P_l^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^1}\sum_{i\in T_l}\left(P_i^t(k)\right)^{-1}\right]^{-1}\\ &= \left[\sum_{i\in T_j}\left(P_i^t(k)\right)^{-1}\right]^{-1}. \end{aligned}$$

Third, we consider the node j is a three-relay node. We assume $V_{j,c}^2$ is the j's children set in which the elements are all the two-relay nodes. Similarly, according to Theorem 1 and the above proof, we have

$$\begin{aligned} &P_j(k)\\ &= \left[\left(P_j^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^{leaf}} \left(P_l^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^1}\left(P_l^t(k)\right)^{-1}\right.\\ &\qquad \left. + \sum_{l\in V_{j,c}^2}\left(P_l^t(k)\right)^{-1}\right]^{-1}\\ &= \left[\left(P_j^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^{leaf}} \left(P_l^t(k)\right)^{-1} + \sum_{l\in V_{j,c}^1}\sum_{i\in T_l}\left(P_i^t(k)\right)^{-1}\right.\\ &\qquad \left. + \sum_{l\in V_{j,c}^2}\sum_{i\in T_l}\left(P_i^t(k)\right)^{-1}\right]^{-1}\\ &= \left[\sum_{i\in T_j}\left(P_i^t(k)\right)^{-1}\right]^{-1}. \end{aligned}$$

Since the sink is a three-relay node obviously, we obtain $P_n(k) = [\sum_{i\in T_n}(P_i^t(k))^{-1}]^{-1}$. In addition, T_n is actually V_1. Therefore, the theorem has been proved. □

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 Sections 3 and 4, $P_n^{V_1}(k)$ and $P_n^{V_2}(k)$ are the estimate error covariance matrices of the sink v_n for the lth 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 lth scheduling period. By using our estimate aggregation approach, if the estimation information can be aggregated to v_n within one scheduling period, we have

$$P_n^{V_1}(k) = \left[\sum_{v_i\in V_1}\left(P_i^t(k)\right)^{-1}\right]^{-1} > \left[\sum_{v_i\in V_1^1}\left(P_i^t(k)\right)^{-1}\right]^{-1} = P_n^{V_1^1}(k)$$

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 $P_1^p(k)$ ($\geq$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 the 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

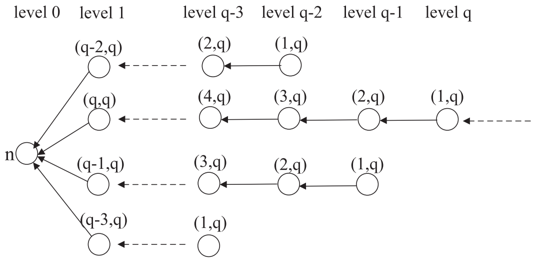


Fig. 4. An upper bound example based on our estimate aggregation approach.

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 state estimates of time k can be aggregated to the sink by any method before time $k+q$.

Theorem 5. *Suppose that n_i is the number of the nodes which are at the ith 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 the nodes whose state estimates 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.*

Proof. We prove by construction. In Fig. 4, we construct a tree network example in which the root has four children nodes and all the other nodes have one child at most. There is only one node in the level q, two nodes in the level $q-1$, three nodes in the level $q-2$, and four nodes in the level $q-3$. The distance between any pair of nodes in the network is greater than r_I except that between a child-parent pair. By using our estimate aggregation approach, every node obtains its own schedule time expressed by (x, y) beside the node in Fig. 4 where x and y are, respectively, the $TSFT$ and NSP of the node. From the schedule, we can see that all the estimates of the nodes whose level is less than or equal to q can be aggregated to the sink within the delay constraint q. This finishes the proof. □

Theorem 7. *Under the protocol interference model, the number of the nodes whose state estimates 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.

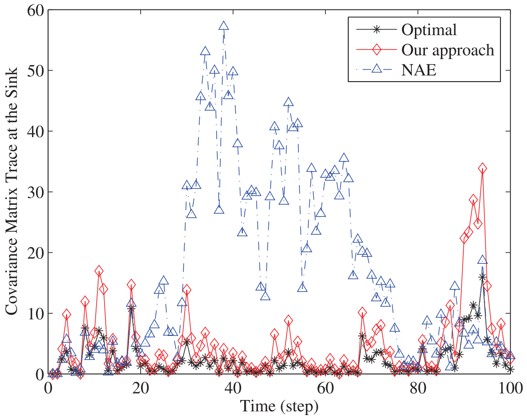


Fig. 5. Covariance matrix trace at the sink for three methods.

6 Simulation Analysis

We implemented the proposed in-network estimation approach and conducted the simulation using MATLAB 7.0. In our simulation, we randomly deploy the sensors into a region of $200\ \mathrm{m} \times 200\ \mathrm{m}$. 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 [27], 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:

$$B^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad B^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad B^3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Assume the maximum level number of the aggregation tree 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 nonaggregation 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 the relays. For satisfying the delay constraint, the estimates are gathered by the sink level by level starting from the lowest level.

First, we evaluate the overall efficiency of our in-network estimation approach. The number of nodes is 500. All sensors have the same transmission radius which is fixed to 25 m. The delay constraint q is 100 time slots. Fig. 5 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

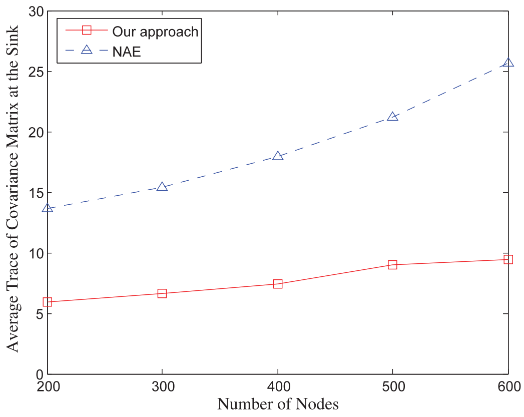


Fig. 6. Average trace of covariance matrices at the sink with different number of nodes.

performance to the optimal method. This is because that our in-network estimation approach has the ability to compensate for the estimate 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 so 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.

Second, we compare the estimation accuracy of our in-network estimation approach with that of NAE under the different network parameters. In Fig. 6, the transmission radius of each sensor node is fixed to 25 m. The figure shows the average trace of the covariance matrices at the sink by running the two algorithms while the number of nodes increases. In Fig. 7, the number of nodes is fixed to 500. The figure compares the average trace of the covariance matrices at the sink using the two algorithms when the transmission radius varies. It can be seen from the two figures that the average estimation accuracy of our in-network estimation approach outperforms that of NAE significantly. Moreover, the estimation performance improvement will be larger with the increase of the number of the nodes and the transmission radius. This is because that the number of the estimates gathered by the sink decreases when the opportunity of the wireless interference increases.

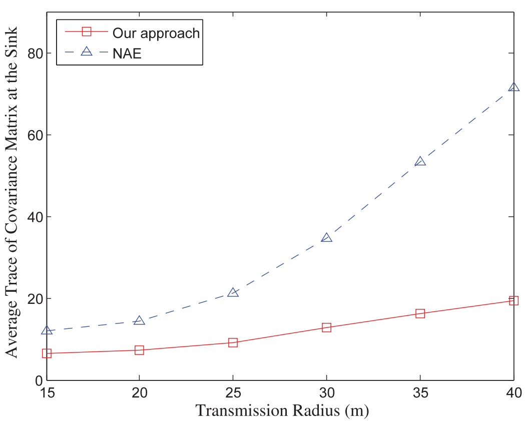


Fig. 7. Average trace of covariance matrices at the sink with different transmission radius.

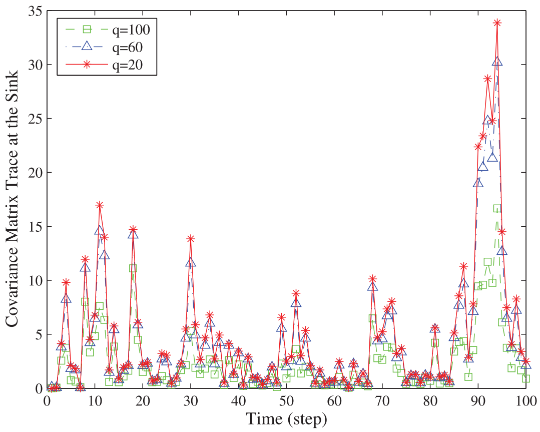


Fig. 8. Covariance matrix trace at the sink for our in-network estimation approach.

Third, Figs. 8 and 9 show the traces of the estimate error covariance matrix at the sink with every estimation time step for our in-network estimation approach and NAE, respectively. From Fig. 8, 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. 9, 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.

7 Related Work

7.1 State Estimation Using WSNs

Estimation over lossy networks has been well studied in recent years. Sinopoli et al. [11] considered the problem of performing Kalman filtering with intermittent observations whose arrival is modeled as a random process. Smith and Seiler [12] proposed a suboptimal but computationally efficient estimator that can be applied when the arrival process is modeled as a Markov chain. In [13] and [14], the authors proposed to estimate the process state (or encode the sensor measurements) at the sensor side of the link

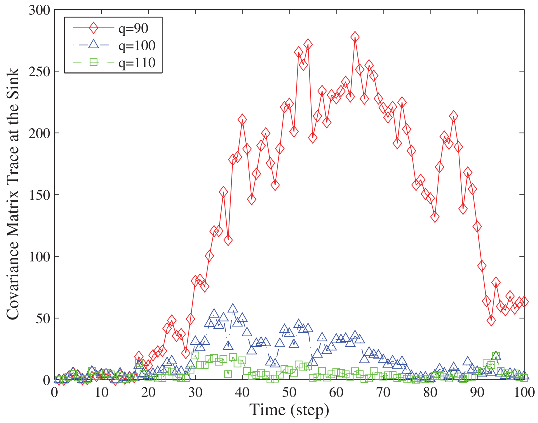


Fig. 9. Covariance matrix trace at the sink for NAE.

without assuming any statistical model for the data loss process. Schenato et al. [15] 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. [31] 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 and Bekey [16] decomposed a single Kalman filter into a number of smaller communicating filters for the multirobot localization problem. Sun and Deng [17] proposed a multisensor optimal information fusion criterion weighted by matrices in the linear minimum variance sense. Based on this optimal fusion criterion, a general multisensor 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. [18] presented a distributed Kalman smoother state estimator. Three novel distributed Kalman filtering algorithms for sensor networks were proposed in [19]. Speranzon et al. [20] proposed a new distributed algorithm for cooperative estimation of a slowly time-varying signal using a wireless sensor network. 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 cannot be guaranteed within any given time window.

7.2 Delay-Efficient Scheduling for Data Aggregation

Data aggregation is considered to be an effective method for improving the 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 the network. Consequently, the data aggregation delay from the network to a distinguished sink and the energy consumption are reduced because the data needed to be scheduled in the network is reduced [21], [22].

Minimum delay data aggregation in WSNs under various interference models has been proven to be NP-hard [24], and several approximation algorithms have been proposed recently, e.g., [24], [25], [27], [26], [28]. Chen et al. [24] 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. [25] 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. [26] 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 [27], and the delay is at most $24D + 6\Delta + 16$ time slots where D is the network diameter. Xu et al. [28] 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 the 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.

In [15] and [35], the authors suggested jointly addressing estimation and communication problems in NCSs, but they did not give any practical solutions. In [36] and [37], the authors jointly addressed the estimation and communication problems in CPSs, and designed an adaptive-reliability transport protocol. But they did not consider the real-time issue of control systems.

8 Conclusions and Future Directions

In this paper, we jointly addressed the state estimation and scheduling problems in multihop WSNs. For accurately estimating a process state as well as satisfying rigid delay constraints, we proposed 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 cannot 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. 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.

CPSs have been attracting significant attention in the past few years. Though the research of CPSs can benefit from the related theoretical and technical results of communication, control, embedded computing, etc., simply integrating the various techniques to build CPSs is usually not efficient or even not feasible. It is because that the optimization of every component or subsystem of a system does not insure the optimization of the whole system. A more desirable method is to jointly design or optimize the different parts of the networked system. This paper gives an initial attempt to codesign the estimation and transmission methods over cyber-physical networks, and the further efforts are still required.

First, as one of the future research directions, the duty cycle wireless networks should be considered. For saving the limited energy of sensor nodes, the duty cycle design of WSNs is prevalent in practice. Therefore, how to tradeoff the energy consumption and the control performance is an urgent issue in CPSs. Second, the cyber-physical network usually supports the multiple control tasks which have the different QoS demands, and thus another problem is how to effectively schedule the different kinds of feedback data for meeting the different system performance demands in the network. Third, this paper only considers the centralized control scheme in which the sensor data are gathered to one sink, but the distributed control is more appropriate in some scenarios. The cyber-physical codesign of wireless networking and control for distributed control systems is still open.

ACKNOWLEDGMENTS

The research of Haitao Zhang and Huadong Ma is supported by National Basic Research Program of China (973 Program) under grant No. 2011CB302701; National Natural Science Foundation of China under Grant Nos. 60833009 and 61133015; China National Funds for Distinguished Young Scientists under Grant No. 60925010; the Program for Changijiang Scholars and Innovative Research Team in University under Grant No. IRT1049. The research of Xiang-Yang Li is partially supported by the US National Science Foundation (NSF) CNS-0832120; NSF CNS-1035894; National Natural Science Foundation of China under Grant No. 61170216; Program for Zhejiang Provincial Key Innovative Research Team; Program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program). The preliminary result [3] was published at ACM/IEEE ICCPS 2011.

REFERENCES

[1] R. Poovendran, "Cyber-Physical Systems: Close Encounters between Two Parallel Worlds," *Proc. IEEE,* vol. 98, no. 8, pp. 1363-1366, Aug. 2010.

[2] E.A. Lee, "Cyber Physical Systems: Design Challenges," *Proc. IEEE Int'l Symp. Object Oriented Real Time Distributed Computing (ISORC '08),* pp. 363-369, May 2008.

[3] H. Zhang, H. Ma, and X.-Y. Li, "Estimate Aggregation with Delay Constraints in Multihop Wireless Sensor Networks," *Proc. IEEE/ACM Second Int'l Conf. Cyber-Physical Systems (ICCPS '11),* pp. 184-193, Apr. 2011.

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

[5] V. Singhvi, A. Krause, C. Guestrin, J.H. Garrett Jr., and H.S. Matthews, "Intelligent Light Control Using Sensor Networks," *Proc. Third Int'l Conf. Embedded Networked Sensor Systems (SenSys '05),* Nov. 2005.

[6] M. Kintner-Meyer and R. Conant, "Opportunities of Wireless Sensors and Controls for Building Operation," *Energy Eng. J.,* vol. 102, no. 5, pp. 27-48, 2005.

[7] http://www.cs.iit.edu/~winet/CPS2011/index.html, 2012.

[8] L. Mottola and G.P. Picco, "Programming Wireless Sensor Networks with Logical Neighborhoods: A Road Tunnel Use Case," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '07),* Nov. 2007.

[9] P. Costa, G. Coulson, R. Gold, M. Lad, C. Mascolo, L. Mottola, G.P. Picco, T. Sivaharan, N. Weerasinghe, and S. Zachariadis, "The RUNES Middleware for Networked Embedded Systems and Its Application in a Disaster Management Scenario," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07),* pp. 69-78, Mar. 2007.

[10] 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.

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

[12] S. Smith and P. Seiler, "Estimation with Lossy Measuraments: Jump Estimators for Jump Systems," *IEEE Trans. Automatic Control,* vol. 48, no. 12, pp. 1453-1464, Dec. 2003.

[13] Y. Xu and J. Hespanha, "Estimation under Controlled and Uncontrolled Communications in Networked Control Systems," *Proc. 44th IEEE Conf. Decision and Control European Control Conf. (CDC-ECC),* pp. 842-847, Dec. 2005.

[14] 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.

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

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

[17] S.-L. Sun and Z.-L. Deng, "Multi-Sensor Optimal Information Fusion Kalman Filter," *Automatica,* vol. 40, no. 6, pp. 1017-1023, June 2004.

[18] I.D. Schizas, G.B. Giannakis, S.I. Roumeliotis, and A. Ribeiro, "Consensus in Ad Hoc WSNs with Noisy Links Part II: Distributed Estimation and Smoothing of Random Signals," *IEEE Trans. Signal Processing,* vol. 56, no. 4, pp. 1650-1666, Apr. 2008.

[19] R. Olfati-Saber, "Distributed Kalman Filtering for Sensor Networks," *Proc. 46th IEEE Conf. Decision and Control,* pp. 5492-5498, Dec. 2007.

[20] A. Speranzon, C. Fischione, and K.H. Johansson, "Distributed and Collaborative Estimation over Wireless Sensor Networks," *Proc. 45th IEEE Conf. Decision and Control,* pp. 1025-1030, Dec. 2006.

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

[22] M.A. Sharaf, J. Beaver, A. Labrinidis, and P.K. Chrysanthis, "Tina: A Scheme for Temporal Coherency-Aware In-Network Aggregation," *Proc. ACM Int'l Workshop Data Eng. for Wireless and Mobile Access (MobiDE),* pp. 69-76, Sept. 2003.

[23] H. Luo, H. Tao, H. Ma, and S.K. Das, "Data Fusion with Desired Reliability in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 22, no. 3, pp. 501-513, Mar. 2011.

[24] X. Chen, X. Hu, and J. Zhu, "Minimum Data Aggregation Time Problem in Wireless Sensor Networks," *Proc. First Int'l Conf. Mobile Ad-Hoc and Sensor Network,* pp. 133-142, May 2005.

[25] 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," *Proc. IEEE INFOCOM,* pp. 366-372, May 2007.

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

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

[28] X. Xu, X.-Y. Li, X. Mao, S. Tang, and S. Wang, "A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 22, no. 1, pp. 163-175, Jan. 2011.

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

[30] P.-J. Wan, K.M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM,* pp. 1597-1603, June 2002.

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

[32] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Technical Report TR95-041, UNC-Chapel Hill, http://www.cs.unc.edu, Nov. 2000.

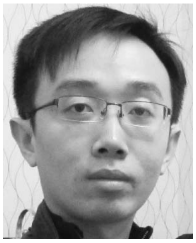
[33] N.A. Carlson, "Federated Square Root Filter for Decentralized Parallel Processes," *IEEE Trans. Aerospace and Electronic Systems,* vol. 26, no. 3, pp. 517-525, May 1990.

[34] S. Sun, "Multi-Sensor Optimal Information Fusion Kalman Filter for Discrete Multichannel ARMA Signals," *Proc. IEEE Int'l Symp. Intelligent Control,* pp. 377-382, Oct. 2003.

[35] J. Baillieul and P.J. Antsaklis, "Control and Communication Challenges in Networked Real-Time Systems," *Proc. IEEE,* vol. 95, no. 1, pp. 9-28, Jan. 2007.

[36] H. Ahmadi and T. Abdelzaher, "An Adaptive-Reliability Cyber-Physical Transport Protocol for Spatiotemporal Data," *Proc. 30th IEEE Real-Time Systems Symp. (RTSS '09),* pp. 238-247, Dec. 2009.

[37] H. Ahmadi, T. Abdelzaher, and I. Gupta, "Congestion Control for Spatio-Temporal Data in Cyber-Physical Systems," *Proc. ACM/IEEE Int'l Conf. Cyber-Physical Systems (ICCPS '10),* pp. 89-98, Apr. 2010.



Haitao Zhang received the BS degree in mathematics from the Dalian University of Technology, China in 2006, and the MS degree in computer science from Northeastern University, Shenyang, China, in 2008. He is currently working toward the PhD degree in the School of Computer Science, Beijing University of Posts and Telecommunications, China. From 2010 to 2011, he was a visiting PhD student at the Department of Computer Science, Illinois Institute of Technology, Chicago. His research interests include Cyber-Physical Systems, Wireless Sensor Networks, Internet of Things, and video processing.

Huadong Ma received the BS degree in mathematics from Henan Normal University in 1984, the MS degree in computer science from the Shenyang Institute of Computing Technology, Chinese Academy of Science in 1990, and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Science in 1995. He is currently a professor and the director of the Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, dean of the School of Computer Science, Beijing University of Posts and Telecommunications, China. He visited UNU/IIST as a research fellow in 1998 and 1999, respectively. From 1999 to 2000, he held a visiting position in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. He was a visiting professor at The University of Texas at Arlington from July to September 2004, and a visiting professor at the Hong Kong University of Science and Technology from December 2006 to February 2007. His current research focuses on multimedia system and networking, Internet of things and sensor networks, and he has published more than 100 papers and four books on these fields. He is a member of the IEEE and the ACM.

Xiang-Yang Li received the BEng degree in computer science and bachelor's degree in business management from Tsinghua University, P.R. China in 1995 and the MS and PhD degrees in computer science from the University of Illinois at Urbana-Champaign, in 2000 and 2001. He has been an associate professor since 2006 and an assistant professor of computer science at the Illinois Institute of Technology from 2000 to 2006. His research interests include cyber-physical systems, wireless sensor networks, social networks, and security. He is an editor of *IEEE TPDS*, *IEEE TMC*, *Networks: An International Journal*, *Ad Hoc & Sensor Wireless Networks: An International Journal*, and was a guest editor of special issues for several journals, including *IEEE Journal on Selected Areas in Communications*, and *ACM Mobile Networks and Applications*. He coedited several books, including, *Encyclopedia of Algorithms*, published in 2008. In 2008, he published a monograph *Wireless Ad Hoc and Sensor Networks: Theory and Applications*, by Cambridge University Press. He is a senior member of the IEEE and a member of the ACM.

Shaojie Tang received the BS degree in radio engineering, Southeast University, China, 2006. He is currently working toward the PhD degree in computer science at the Illinois Institute of Technology. His research field is on algorithm design, optimization, security of wireless networks, electronic commerce as well as online social network. He is a student member of the IEEE.