

Optimization of Web Service-Based Data-Collection System With Smart Sensor Nodes for Balance Between Network Traffic and Sensing Accuracy

Chen Hou^{ID}, Qianchuan Zhao^{ID}, *Senior Member, IEEE*, and Tamer Başar^{ID}, *Life Fellow, IEEE*

Abstract—Web services integrate various components in the Internet of Things (IoT). In a Web service-based data-collection system with multiple smart sensor nodes periodically sampling and estimating the same unknown physical parameter of interest, the smart sensor nodes first submit their estimates to the Web server, and then, the server picking the one with the minimum error seems to be a practical way to arrive at a minimum error estimate (MEE). More submissions provide the Web server with more candidates to consider, which can maximize the probability of the server guaranteeing the MEE, while also leading to more network traffic. Therefore, how to make the optimal tradeoff between network traffic and sensing accuracy arises as an interesting problem. This article proposes a network traffic-dependent probability threshold policy within an intended underlying optimization-theoretical framework to address this problem. The policy is such that the smart sensor nodes submit their estimates and corresponding estimation errors (ECEEs) to the Web server within a tolerable network traffic threshold while maximizing the probability of the server delivering the MEE. Theoretical analysis, simulation, and field experiments document and illustrate its performance.

Note to Practitioners—This article addresses the interesting tradeoff between sensing accuracy and network traffic demand in the Web service-based data-collection system that operates in some remote areas with limited network traffic. It helps to improve the operation efficiency of the Internet-of-Things (IoT) systems that employ Web service technology to enable the Web server to deliver minimum error estimate with maximum probability while keeping the network traffic within a given range. Our simulation and experimental investigations show that the solution developed here outperforms existing solutions.

Manuscript received January 1, 2020; revised April 5, 2020 and July 20, 2020; accepted October 5, 2020. Date of publication October 29, 2020; date of current version October 6, 2021. This article was recommended for publication by Associate Editor K. Zhu and Editor B. Vogel-Heuser upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Project of China under Grant 2017YFC0704100 and Grant 2016YFB0901900, in part by the National Natural Science Foundation of China under Grant 61425027, in part by the 111 International Collaboration Program of China under Grant BP2018006, and in part by the BNRist Program under Grant BNR2019TD01009. (Corresponding author: Qianchuan Zhao.)

Chen Hou and Qianchuan Zhao are with the Center for Intelligent and Networked Systems (CFINS), Department of Automation and BNRist, Tsinghua University, Beijing 100084, China (e-mail: houc12th@163.com; zhaocq@tsinghua.edu.cn).

Tamer Başar is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: basar1@illinois.edu).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TASE.2020.3030835

Index Terms—Internet of Things (IoT), multiple smart sensor nodes, network traffic, sensing accuracy, web service.

NOMENCLATURE

| | |
|-------------------|---|
| MEE | Minimum error estimate. |
| ECEE | Estimate and corresponding estimation error. |
| s_i | i th smart sensor node |
| n | Number of all smart sensor nodes. |
| \mathbb{N} | $= \{s_1, s_2, \dots, s_n\}$, the set of all smart sensor nodes. |
| Δ_1 | An action, submit the estimate to the Web server. |
| Δ_2 | An action, do not submit the estimate to the Web server. |
| \mathbb{A}_i | $= \{\Delta_1, \Delta_2\}$, the set of all actions of s_i . |
| a_i | Action taken by s_i . |
| \mathbf{a}_{-i} | $= (a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$, the vector of actions taken by the opponents of s_i . |
| U_i | $= U_i(a_i)$, the payoff of s_i . |
| π | $= (P\{a_1 = \Delta_1\}, P\{a_1 = \Delta_2\}, \dots, P\{a_n = \Delta_n\})$, the policy. |
| $P\{X\}$ | Probability that event X occurs. |
| G | $= \underbrace{[0 \ 1] \times [0 \ 1] \times \dots \times [0 \ 1]}_n$, the policy space of π . |
| P_π | Cumulative probability under π . |
| C_π | Cumulative network traffic consumed under π . |
| Q | Available network traffic in a periodic cycle. |
| e_i | Estimation error of s_i . |
| δ_i | indicates whether the sever obtains the MEE by s_i taking a_i . |
| $C_i(a_i)$ | Network traffic consumed by s_i taking a_i . |
| c | Average network traffic consumed by an estimate-submission without data block. |
| ε | Average variation of network traffic consumed by an estimate-submission per additional estimate submitted to the Web server. |
| λ_i | represents that s_i is the i th node submitting the estimate. |
| π^* | Optimal policy in G , it is the optimal policy for each sensor nodes to determine whether to submit its estimate to the Web server in order to maximize the probability of obtaining the minimum error estimate while maintaining |

- the server network traffic at the required level.
- r Number of smart sensor nodes that must take Δ_1 .
- $P(r) = P_\pi$ ($P(r)$ is used in the Appendix).
- $C(r) = C_\pi$ ($C(r)$ is used in the Appendix).
- $r^* = \arg \max_{r \in \{1, 2, \dots, n\}} P(r)$, the optimal r , the number of sensor nodes that must submit their estimates to the server under the optimal policy.

I. INTRODUCTION

WITH the capability to enable communication between machines via the Worldwide Web, Web services have infiltrated into various Internet-of-Things (IoT) applications to bridge different components, such as Web server and smart sensor nodes [1]–[4], where the latter differs from the traditional sensor nodes by having the intelligence to make decisions [5].

The structure of a Web service-based data-collection system with multiple smart sensor nodes is shown in Fig. 1, in which the nodes: 1) are independent of, and cannot communicate with, each other; 2) periodically sample and estimate the same unknown physical parameter of interest; 3) obtain their ECEEs¹ and submit the ECEEs to the Web server according to some criteria; and 4) have no knowledge of each other’s ECEE.

The Web server works according to the following mechanism [6], [7]. To initiate a transmission control protocol (TCP) connection with the Web client, the Web server must listen on a network port for the client request. If the TCP connection is successfully built, and the Web client request is made, then a response with several hundred bytes must be issued from the Web server to the client no matter whether this request is successful or not. If successful, then a data packet with the required information will be given; otherwise, a detailed reason for the failure will be included in the response.

The smart sensor nodes do not know and cannot obtain each other’s estimates, so, for the Web server to obtain the MEE, all the nodes first submit their estimates to the Web server, and then, the server picking the one with the minimum error seems to be a practical method [8]. However, each submission has to consume network traffic, and the network resource may be too limited to support all the submissions. Therefore, new policies are needed to balance the network traffic and sensing accuracy under the environments where the available network resource is not sufficient to support each node to issue a data packet to the Web server. Section II provides a review of the related literature.²

¹The rest of this article will not differentiate between “estimate” and “ECEE.”

²The readers may wonder why not letting the smart sensor nodes obtain information on others’ estimates by requesting from the Web server. The abovementioned mechanism shows that the Web server cannot actively send this information to the smart sensor nodes because the latter plays the role of a Web client. Whenever this information is requested by the smart sensor node, it has to initiate a request to and receive a response from the Web server, which leads to network traffic cost [7]. Thus, this is infeasible under the environments where the average network resource is not sufficient to support each smart sensor node to issue a data packet to the Web server.

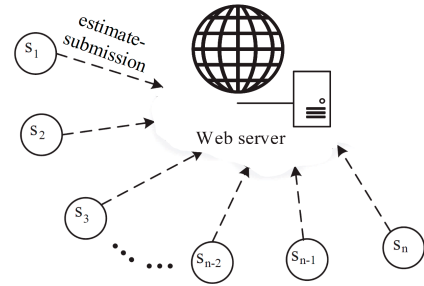


Fig. 1. Structure of a Web service-based data-collection system with multiple smart sensor nodes (for each sensor node, its estimate and corresponding estimation error are private to it before reaching the Web server).

To achieve the abovementioned balance, this article attempts to find an effective policy to maximize the probability of the Web server obtaining the MEE with the cumulative network traffic consumed being no larger than a given threshold based on six conditions as listed in the follows.

- 1) Each smart sensor node can obtain an estimate which is private to the node before reaching the Web server.
- 2) The sensor nodes have no knowledge of each other’s estimates and whether the other nodes have submitted their estimates or not and cannot obtain this information from the Web server or by mutual communication.
- 3) To let the Web server obtain the MEE, some (at least one) estimate-submissions from the nodes to the Web server must happen, and the MEE will be selected by the Web server from among all the estimates submitted (the Web server can recognize the relative MEE).
- 4) Each smart sensor node will submit its estimate only if it considers that it has a smaller estimation error than all the previous ones that have submitted their estimates;
- 5) Each smart sensor node has a node number which is arbitrary and exclusive (it is private to the node).
- 6) The other dependable source, such as power, is sufficient for each sensor node to make decision and submission.

Accordingly, we have made in this article two contributions as follows.

- 1) An optimization-theoretical framework is established for the Web service-based data-collection system with multiple smart sensor nodes.
- 2) We develop a network traffic-dependent probability threshold policy for the smart sensor nodes to determine whether to submit their estimates to the Web server in order to maximize the probability of the server obtaining the MEE while maintaining the required level of network traffic.

The rest of this article is organized as follows. We will review the relevant literature in Section II, state and formulate the problem in Section III, present the algorithms in Section IV, carry out algorithm analysis in Section V, perform experiments in Section VI, and draw conclusions in Section VII.

II. LITERATURE REVIEW

Web services have been employed into sensor networks, such as a sink node, but with more powerful

functions [9]–[12]. However, resource constraints imposed by practical application conditions often prohibit the use of Web service technologies [13], [14]. To overcome this difficulty, the mutual communication of smart sensor nodes has been exploited to optimally allocate limited resources to multiple nodes [15]. For example, when the network traffic is limited, some smart sensor nodes are selected as proxies, and they first receive the information from the other nodes and then filter the dummy information that wastes network traffic [16]. However, mutual communication between smart sensor nodes is not allowed everywhere [17], which drives us to focus on the design of strategies for smart sensor nodes to interact with the Web server on the condition that the network resource is constrained, and the sensor nodes cannot exchange information.

The Web service-based data-collection system with multiple smart sensor nodes aims to increase the level of accuracy of estimations of the physical parameters of interests. Receiving more estimates from the smart sensor nodes allows the Web server to report with greater sensing accuracy. However, this creates challenges when the network traffic available to smart sensor nodes is not at the level to support each estimate-submission [18]. Hence, many advanced studies have been carried out over the past decades. Based on the structure of the methods, we roughly divide them into three classes: probability-dependent threshold algorithm (PDTA) [19], network traffic exhaustive algorithm (NTEA) [20], and web service request-assistant algorithm (WSRA) [21].

PDTA commands the smart sensor nodes to submit their estimates to the Web server with a fixed probability [19]. For example, Beyah and Cai [22] proposed a probabilistic network model in which the data of smart sensor nodes is transmitted probabilistically. Liu *et al.* [23] developed a data-collection scheme for the sensor networks to conduct data transmission probabilistically. NTEA requires the network traffic to support as many estimate-submissions as possible until exhausted [20]. For example, Kyusakov *et al.* [24] embedded simple object access protocol (SOAP)-based Web services into smart sensor nodes to enable them to use as much network traffic as possible. Glombitza *et al.* [25] implemented Web services on smart sensor nodes without restrictions on the use of network traffic supplied. WSRA lets the smart sensor nodes acquire knowledge of the relative MEE by requesting from the Web server [21]. For example, Han *et al.* [26] made use of the service-oriented architecture (SOA) to establish a building automation system with multiple smart sensor nodes that capture the needed information from the Web server. Khan *et al.* [27] comprehensively classified the mobility sink-dependent data collection and dissemination policies into three policies in all of which the mobility sink plays the role of the Web server and provides the needed information for the other smart sensor nodes. However, PDTA does not ensure that the server can obtain at least one estimate; NTEA always leads to waste of network traffic, and WSRA's feedback from the server to the smart sensor nodes must consume network traffic [28], [29]. Therefore, an effective strategy in practice to make the optimal tradeoff between network traffic

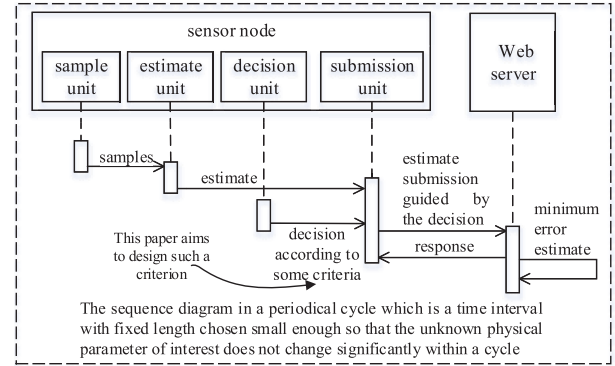


Fig. 2. Working time sequence of smart sensor nodes as the form of sequence diagram.

and sensing accuracy is called for, with the available network traffic constrained.

To balance the network traffic and sensing accuracy in the network traffic-constrained environment for the Web service-based data collection system in which the smart sensing nodes cannot exchange information, the noncooperative game framework seems to be a feasible one in which each smart sensor node wants to maximize its utility function, as researched in our previous work [30]. However, what is solved in this article is the probability of the Web server achieving the MEE at the required level of network traffic. It is the sum of utility functions of all smart sensor nodes and a single criterion optimization problem and not the maximization of individual utility functions as in the standard noncooperative game with the Nash equilibrium [31]. Therefore, the noncooperative game aspect has to be eliminated. Instead, the intended underlying optimization framework aiming at a single criterion optimization problem [32] is adopted. Accordingly, the main contribution is the formulation of the problem of smart sensor nodes determining whether to submit their estimates to the Web server or not, under network traffic constraints, as an optimization problem, and further develop a technique to capture the optimal tradeoff between the network traffic and sensing accuracy.

III. PROBLEM STATEMENT AND FORMULATION

A. Problem Statement and Main Challenges

The data-collection system in Fig. 1 is working in a periodic way, as shown in Fig. 2: at the beginning of a time interval with fixed length (we call it a periodic cycle and assume that it is chosen small enough so that the unknown physical parameter of interest does not change significantly within a cycle): 1) the smart sensor nodes start to sample and estimate the unknown physical parameter; 2) if the periodic cycle ends, then the nodes submit their estimates to the Web server according to some criterion³; and 3) the server picks the estimate with minimum error from all the estimates submitted and sees it as the MEE. The procedure repeats in the next cycle [33].

Each smart sensor node can obtain an estimate, but does not know and cannot obtain others' estimates. The estimates

³This article aims to design such a criterion.

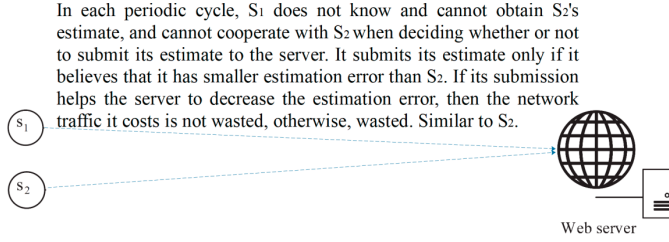


Fig. 3. Process of two smart sensor nodes making decisions.

in different sensor nodes differ from each other,⁴ and can successfully reach the Web server if they are submitted. The way that all the sensor nodes submitting their estimates to the Web server and then the server picking the one with minimum error leads to unnecessary network traffic. For example, for the two sensor nodes s_1 and s_2 , as shown in Fig. 3, if s_2 has a smaller estimation error than s_1 and has submitted its estimate to the Web server, then it is unnecessary for s_1 to submit its because its estimate cannot help the Web server to decrease the estimation error, while its estimate-submission has to consume network traffic. Unfortunately, s_1 has no knowledge of s_2 's estimate, as well as whether s_2 has submitted its estimate or not, and cannot acquire this information by asking s_2 or requesting from the Web server, because s_2 and s_1 cannot mutually communicate, and the Web server mechanism shows that the server cannot actively send this information to s_1 . Therefore, s_1 cannot compare its estimate with s_2 's before making the decision. With this in mind, the problem this article would like to solve can be stated as follows, with the precise formulation given in Section III-B as an optimization problem.

Problem [Estimate-Submission Decision Problem (ESDP)]: To design a policy for each smart sensor node to determine whether to submit its estimate to the Web server in order to maximize the probability of the server obtaining the MEE while maintaining the network traffic at the required level.

Accordingly, the process of ESDP can be described as follows: in each periodic cycle, when the data-sampling is stopped, each smart sensor node obtains an estimate and submits its estimate only if it believes that it has a smaller estimation error than the others submitting their estimates ahead of it.

The main challenges lie in that the smart sensor nodes have no information on others' estimates, as well as whether these estimates have been submitted to the Web server and cannot obtain this information from the server or by mutual communication when making its decision.

B. Problem Formulation

Nomenclature lists the notations used in the rest of this article.

We formulate the ESDP as an optimization problem and first introduce a tuple [32]

$$\langle \mathbb{N}, (\mathbb{A}_i)_{i \in \{1, 2, \dots, n\}}, (U_i)_{i \in \{1, 2, \dots, n\}}, (C_i)_{i \in \{1, 2, \dots, n\}} \rangle \quad (1)$$

to describe the estimate-submission decision process. Here, the following holds.

⁴This is true because it is almost impossible for two different smart sensor nodes to have the same estimate even in the same environment [34].

Algorithm 1: NTPTA

Input: $s_i, n, c, \varepsilon, Q, \Delta_1, \Delta_2, \mathbb{N}$;

Output: a_i ;

```

1 foreach  $s_i \in \mathbb{N}$  do
2   if  $i \in \{1, 2, \dots, r^*\}$  then
3      $a_i \leftarrow \Delta_1$ ;
4   else
5     generate a random sample  $\theta$  from uniform
      distribution  $U(0, 1)$ ;
6     if  $0 < \theta \leq \frac{r^*}{i-1}$  then
7        $a_i \leftarrow \Delta_1$ ;
8     else
9        $a_i \leftarrow \Delta_2$ ;
10  $P_{\pi^*} \leftarrow \frac{r^*}{n} + \sum_{i=r^*+1}^n \left(\frac{1}{n}\right) \left(\frac{r^*}{i-1}\right)$  or  $\frac{r^*}{n} - \frac{r^*}{n} \ln \frac{r^*}{n}$ 
    (for large  $n$ )
    
```

- 1) $\mathbb{N} = \{s_1, s_2, \dots, s_n\}$ is the set of all smart sensor nodes.
- 2) $\mathbb{A}_i = \{\Delta_1, \Delta_2\}$ is the set of all actions of $s_i \in \mathbb{N}$ with Δ_1 and Δ_2 denoting, respectively, whether s_i submits or does not submit its estimate to the Web server.
- 3) $U_i = U_i(a_i)$ is the payoff of s_i with $a_i \in \mathbb{A}_i$ being the action taken by s_i and the payoff denoting the probability of the Web server ending up with the MEE when s_i takes action a_i .
- 4) $C_i = C_i(a_i)$ is the network traffic consumed when s_i takes action a_i .

Let π denote the policy defined by

$$\pi = (P\{a_1 = \Delta_1\}, P\{a_2 = \Delta_1\}, \dots, P\{a_n = \Delta_1\}). \quad (2)$$

Then, since $P\{a_i = \Delta_1\} \in [0, 1]$, the policy space G of π can be given by

$$G = \underbrace{[0, 1] \times [0, 1] \times \dots \times [0, 1]}_n. \quad (3)$$

In a periodic cycle, let P_π denote the cumulative probability of the Web server obtaining the MEE under π and C_π denote the cumulative network traffic consumed under π . Now, we are ready to state ESDP as an optimization problem that can be solved by programming as

$$\begin{aligned} & \max_{\pi \in G} P_\pi \\ & \text{s.t. } C_\pi \leq Q \end{aligned} \quad (4)$$

where Q is the available network traffic consumed in a periodic cycle,⁵ which is not at a sufficient level to support all the smart sensor nodes s_1, s_2, \dots, s_n to submit their estimates.

In the following, we will show that the formulation of ESDP in (4) can be written in terms of the parameters in (1).

Denote by e_1, e_2, \dots, e_n the estimation errors of s_1, s_2, \dots, s_n ; then, we have

$$\text{MEE} = \min(e_1, e_2, \dots, e_n) \quad (5)$$

⁵ $nc + ((n+1)/2)\varepsilon$ is an upper bound of Q . It is the minimum level of network traffic that can support all sensor nodes to submit their estimates and always larger than Q .

and, recalling the footnote 3, we have

$$e_x \neq e_y. \quad (6)$$

s_1, s_2, \dots, s_n are identical to each other except their estimates, so we have

$$P\{e_i = \min(e_1, e_2, \dots, e_n)\} = \frac{1}{n}. \quad (7)$$

Let δ_i be the indicator function of whether the Web server achieves the $\min(e_1, e_2, \dots, e_n)$ with s_i taking a_i , that is,

$$\delta_i = \begin{cases} 1, & \min(e_1, e_2, \dots, e_n) \text{ is obtained with } s_i \text{ taking } a_i \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Then, by (7) and the definition of Δ_1 and Δ_2 , we have

$$P\{\delta_i = 1|a_i\} = \begin{cases} P\{e_i = \min(e_1, e_2, \dots, e_n)\} = \frac{1}{n} & a_i = \Delta_1 \\ 0, & a_i = \Delta_2 \end{cases} \quad (9)$$

and further by the definition of $U_i(a_i)$, we have

$$U_i(a_i) = P\{\delta_i = 1|a_i = \Delta_1\}P\{a_i = \Delta_1\} + P\{\delta_i = 1|a_i = \Delta_2\}P\{a_i = \Delta_2\} = \frac{P\{a_i = \Delta_1\}}{n} \quad (10)$$

$$P_\pi = \sum_{i=1}^n U_i(a_i) = \sum_{i=1}^n \frac{P\{a_i = \Delta_1\}}{n}. \quad (11)$$

Let constant c denote the average network traffic consumed by an estimate-submission without data block. More estimate-submissions lead to more data blocks, and the network traffic consumed by an estimate-submission varies under different number of blocks. Let constant ε denote the average value of such variation per additional estimate submitted to the Web server; then

$$C_i(a_i) = (c + \lambda_i \varepsilon)P\{a_i = \Delta_1\} + 0P\{a_i = \Delta_2\} = (c + \lambda_i \varepsilon)P\{a_i = \Delta_1\} \quad (12)$$

$$C_\pi = \sum_{i=1}^n C_i(a_i) = \sum_{i=1}^n (c + \lambda_i \varepsilon)P\{a_i = \Delta_1\} \quad (13)$$

where λ_i denotes that s_i is the i th smart sensor node submitting the estimate.⁶

Therefore, the ESDP in (4) can be formally rewritten as

$$\begin{aligned} \max_{\pi \in G} & \sum_{i=1}^n \frac{P\{a_i = \Delta_1\}}{n} \\ \text{s.t.} & \sum_{i=1}^n (c + \lambda_i \varepsilon)P\{a_i = \Delta_1\} \leq Q. \end{aligned} \quad (14)$$

⁶The conditions 5) and 6) in Section II-A indicate that we can always let s_i be the i th smart sensor node deciding whether to submit the estimate.

IV. ALGORITHMS

A. Our Algorithm: NTPTA

The network traffic-dependent probability threshold algorithm (NTPTA) working $\forall s_i \in \mathbb{N}$ to determine whether to submit its estimate to the Web server or not is formally given in Algorithm 1, where π^* is the optimal policy for each sensor node to determine whether to submit its estimate to the Web server in order to maximize the probability of the server obtaining the MEE while maintaining the network traffic at the required level, and r^* determined by (16) or (17) (for large n) is the number of sensor nodes that must submit their estimates to the server under the optimal policy π^* .

Brief Description of NTPTA: $\forall s_i \in \mathbb{N}$: 1) if r^* is determined, then NTPTA checks whether $i \in \{1, 2, \dots, r^*\}$; 2) if yes, then $a_i = \Delta_1$; 3) otherwise, generates a random sample θ from uniform distribution $U(0, 1)$; 4) checks whether θ lies in $(0, (r^*/(i-1))]^7$; 5) if yes, then s_i takes Δ_1 ; and 6) otherwise, Δ_2 . After all the smart sensor nodes take actions, P_{π^*} is $(r^*/n) + \sum_{i=r^*+1}^n (1/n)(r^*/(i-1))$ or $(r^*/n) - (r^*/n) \ln(r^*/n)$ (for large n).⁸

B. Other Algorithms

We compare NTPTA with three existing popular algorithms as summarized in Section II: PDTA, NTEA, and WSRA in Algorithms 2, 3, and 4, respectively.

Brief Description of PDTA: $\forall s_i \in \mathbb{N}$, PDTA requires s_i to submit its estimate with probability $(2Q/(n(2c + (n+1)\varepsilon)))$.

Brief Description of NTEA: $\forall s_i \in \mathbb{N}$: 1) if $i \in \{1, 2, \dots, \lfloor (2Q/(2c + (n+1)\varepsilon)) \rfloor\}$, then s_i takes Δ_1 and 2) else, Δ_2 . In other words, NTEA lets the smart sensor nodes submit their estimates successively until the remaining network traffic cannot support an estimate-submission.

Brief Description of WSRA: $\forall s_i \in \mathbb{N}$: 1) if $i \in \{1, 2, \dots, \lfloor (Q/(2c + (n+1)\varepsilon)) \rfloor\}$, then s_i requests the server for e_{relative} and takes Δ_1 and Δ_2 under $e_i < e_{\text{relative}}$ and $e_i \geq e_{\text{relative}}$, respectively and 2) else, s_i takes Δ_2 . In other words, the nodes in WSRA can compare their estimation error with that of the relative MEE requested from the server before taking action.

V. ANALYSIS OF ALGORITHM

The analysis consists of: 1) optimality analysis showing that NTPTA makes the optimal tradeoff between network traffic and sensing accuracy and 2) performance analysis showing that NTPTA is better than PDTA, NTEA, and WSRA.

⁷We use a random sample θ obeying uniform distribution $U(0, 1)$ to obtain the probabilities. Especially, we generate θ from uniform distribution $U(0, 1)$ to charge the action of the sensor nodes that just submit their estimates with respective probabilities: for such kind of sensor node, if the random sample θ generated from uniform distribution $U(0, 1)$ lies in the interval $(0, (r^*/(i-1))]$, then the sensor node s_i should submit its estimate, otherwise not.

⁸ $P_{\pi^*} = (r^*/n) + \sum_{i=r^*+1}^n (1/n)(r^*/(i-1))$ is the exact value of P_{π^*} no matter whether the number n of sensor nodes is small or large, while $P_{\pi^*} \approx (r^*/n) - (r^*/n) \ln(r^*/n)$ is the approximate value of P_{π^*} when n is just large (the latter is easier to calculate than the former when n is large), but the approximation error is no larger than 5% for $n \geq 20$. For example, it is 4.25%, 1.71%, 0.85%, and 0.09% for $n = 20, 50, 100,$ and 1000 , respectively.

Algorithm 2: PDTA [19]

Input: $s_i, n, c, \varepsilon, Q, \Delta_1, \Delta_2, \mathbb{N}$;

Output: a_i ;

```

1 foreach  $s_i \in \mathbb{N}$  do
2   generate a random sample  $\theta$  from uniform distribution
    $U(0, 1)$ ;
3   if  $0 < \theta \leq \frac{2Q}{n(2c + (n+1)\varepsilon)}$  then
4      $a_i \leftarrow \Delta_1$ ;
5   else
6      $a_i \leftarrow \Delta_2$ ;

```

Algorithm 3: NTEA [20]

Input: $s_i, n, c, \varepsilon, Q, \Delta_1, \Delta_2, \mathbb{N}$;

Output: a_i ;

```

1 foreach  $s_i \in \mathbb{N}$  do
2   if  $i \in \left\{1, 2, \dots, \left\lfloor \frac{2Q}{2c + (n+1)\varepsilon} \right\rfloor\right\}$  then
3      $a_i \leftarrow \Delta_1$ ;
4   else
5      $a_i \leftarrow \Delta_2$ ;

```

Algorithm 4: WSRA [21]

Input: $s_i, n, c, \varepsilon, Q, \Delta_1, \Delta_2, \mathbb{N}$;

Output: a_i ;

```

1  $e_{relative}$  is the estimation error of the relative MEE
  obtained by server;
2 foreach  $s_i \in \mathbb{N}$  do
3   if  $i \in \left\{1, 2, \dots, \left\lfloor \frac{Q}{2c + (n+1)\varepsilon} \right\rfloor\right\}$  then
4     let  $s_i$  request the Web server for  $e_{relative}$ ;
5     if  $e_i < e_{relative}$  then
6        $a_i \leftarrow \Delta_1$ ;
7     else
8        $a_i \leftarrow \Delta_2$ ;
9   else
10     $a_i \leftarrow \Delta_2$ ;

```

A. Optimality of NTPTA

Under condition 3) in Section II-A, we let r denote the number of smart sensor nodes that must take Δ_1 .

Theorem 1: The optimal policy $\pi^* = (P\{a_1 = \Delta_1\}, P\{a_2 = \Delta_1\}, \dots, P\{a_n = \Delta_1\})$ for (4) or (14) is given by

$$P\{a_i = \Delta_1\} = \begin{cases} 1, & i \in \{1, 2, \dots, r^*\} \\ \frac{r^*}{i-1}, & i \in \{r^* + 1, r^* + 2, \dots, n\} \end{cases} \quad (15)$$

where r^* is determined by

$$r^* = \max \left\{ r \mid \sum_{i=1}^r (c + i\varepsilon) + \sum_{i=r+1}^n \frac{r(c + i\varepsilon)}{i-1} \leq Q, \right. \\ \left. r \in \{1, 2, \dots, n\} \right\} \quad (16)$$

or (for large n)

$$r^* = \max \left\{ r \mid r \left(c + n\varepsilon - \frac{(r-1)\varepsilon}{2} \right) - r(c + \varepsilon) \ln \frac{r}{n} \leq Q, \right. \\ \left. r \in \{1, 2, \dots, n\} \right\} \quad (17)$$

with

$$P_{\pi^*} = \frac{r^*}{n} + \sum_{i=r^*+1}^n \left(\frac{1}{n} \right) \left(\frac{r^*}{i-1} \right) \quad (18)$$

or (for large n)

$$P_{\pi^*} \approx \frac{r^*}{n} - \frac{r^*}{n} \ln \frac{r^*}{n}. \quad (19)$$

Proof: See the Appendix.

Equation (15) provides the structure of the optimal policy π^* , and (16) and (18) show how to obtain r^* and P_{π^*} , respectively. Meanwhile, for large n , (17) and (19) provide the approximate solution, and the approximation error is no larger than 5% for $n \geq 20$ (for example, it is 4.25%, 1.71%, 0.85%, and 0.09% for $n = 20, 50, 100$, and 1000, respectively).

Theorem 1 guides the smart sensor nodes to determine whether to submit their estimates to the Web server or not. Especially, $s_1, s_2, \dots, s_r, s_{r^*+1}, s_{r^*+2}, \dots, s_n$ should submit their estimates with respective probabilities $\underbrace{1, 1, \dots, 1}_r, 1, (r^*/(r^* + 1)), \dots, (r^*/(n - 1))$. Theorem 1 lays the sound foundation for NTPTA.

B. Performance Comparison

Let P_{PDTA} , P_{NTEA} and P_{WSRA} denote, respectively, the cumulative probabilities of the Web server obtaining the MEE by PDTA, NTEA, and WSRA.

Theorem 2: Under the same network traffic, $P_{\pi^*} > P_{PDTA} \geq P_{NTEA} > P_{WSRA}$, where P_{π^*} is determined by Algorithm 1.

Proof: See the Appendix.

Theorem 2 shows that the performance of NTPTA is better than that of PDTA/NTEA/WSRA in terms of maximizing the probability of the Web server obtaining the MEE under the same level of network traffic. This result follows because of the following.

- 1) NTPTA ensures that the Web server can obtain at least one estimate while PDTA does not.
- 2) NTPTA in every periodic cycle makes full use of the network traffic, while NTEA and WSRA both always waste at least $(2cQ/(2c + (n+1)\varepsilon)) - c[(2Q/(2c + (n+1)\varepsilon))] \geq 0$.
- 3) NTPTA does not request from the server while WSRA does.

In summary, NTPTA has a better performance than PDTA, NTEA, and WSRA by not only considering a more realistic model but also avoiding wasting network traffic.

VI. SIMULATION AND FIELD EXPERIMENT VERIFICATION
A. Experimental Setup

In this section, we compare NTPTA against PDTA, NTEA, and WSRA through both simulations on MATLAB and field experiments on the experimental systems, as shown in Fig. 4.

The experimental system consists of smart sensor nodes, hotspot (smartphone), 4G base, operators' network, and Web server (laboratory-based server). The smart sensor nodes with Wi-Fi but without 4G function working in the remote area are connected to the Web server with the help of the hotspot that enables the mutual conversion between Wi-Fi and 4G. Especially, the smart sensor nodes locally connect the hotspot through a shared Wi-Fi network, while the hotspot remotely communicates with the Web server through the 4G network that is provided by the 4G base.

The Web server is implemented on a laboratory-based personal computer (PC) with the working voltage, resolution, CPU, and hard disk being 220-V ac, 1366×768 , Processor Intel Core i3-3220 CPU at 3.30 GHz, and 500 GB, respectively.

The smart sensor node consists of temperature sensor ER-TH-M5, transmission media universal serial bus, electrical logic converter MAX232, microprocessor MSP430, and serial-interface/Wi-Fi converter HLK-RM04. The smart sensor nodes were assigned with a task to sample the temperature every 18 s before stopping sampling in the periodic cycle set as 5 min.

MAX232 [35] and HLK-RM04 [36] are both signal converters with working voltage 5 V, where the former converts the Recommended Standard 232 (RS232) to transistor-transistor logic (TTL) signal, while the latter converts the TTL to transmission control protocol (TCP) data package. Here, we supplied 5-V voltage for them and the universal serial bus.

As a mixed-signal processor, MSP430 can work under the voltage from 1.8 to 3.3 V [37], and we supplied 3.3 V here. It can work under the 0–25 MHz frequency, and we supplied 4 MHz here. By its P3.4 and P3.5 pins, it can send the commands to and receive the samples from ER-TH-M5, respectively, through the universal serial bus and MAX232 with Baud Rate 9600 b/s. By its P3.6 and P3.7 pins, it can send the samples to and receive the Web service response from the Web server, respectively, through the universal serial bus and HLK-RM04 with a baud rate of 9600 b/s. It makes use of its timer, Timer A (TA), to realize the duration of 18 s and 5 min.

As a digital sensor to sample the temperature, ER-TH-M5 can work under the voltage from 7 to 25 V [38], and we supplied 10 V here. It can work under the 4-MHz frequency, matching MSP430's, so we supplied 4 MHz here. By the TXD/D+ and RXD/D− pins, it can send the samples to and receive commands from MSP430, respectively, through the universal serial bus and MAX232 with Baud Rate 9600 b/s.

To calibrate all the sensors, we first selected the sensors having the same type, i.e., all the temperature sensors are ER-TH-M5, and then resumed them to default settings. The noise produced in the temperature measurement is the thermal noise of the sensor. It obeys normal distribution with mean value 0 and variance 0.5^2 given by the temperature sensor datasheet.⁹ Its spectrum is a uniform distribution with a magnitude of 0.5^2 .

The Wi-Fi network uses the band of 2.4G and the protocol of IEEE 802.11 b. The channel capacity available to a single smart sensor node approximates 1000 B. When all smart

sensor nodes are operating, the channel capacity available to each sensor node is approximately between 1000 and 1300 B, but totaling to less than the available network traffic Q .

We constrained the channel capacity by the built-in function “data limit” of the hotspot (smartphone). It allows us to set the channel capacity limit. When the limit is reached, the hotspot will be automatically disabled, the Wi-Fi network will disappear, and the connection of the smart sensor nodes to the Web server will be terminated. This means that the channel capacity constraint occurs in the hotspot.

The smart sensor nodes work periodically as stated in Section I, with the periodic cycle of 5 min. We obtained the MEE by the following steps: In a periodic cycle: 1) each smart sensor node iteratively calculates its estimate and corresponding estimation error by the first and second equations, respectively, in (20), which is proposed in our previous work [33] (see (15) in [33]), where, at time t , z_t is the sample, $\sigma_{z,t}^2$ is the sample variance, $\hat{\mu}_t$ is the Bayesian posterior estimate ($\hat{\mu}_t$ is initialized as the first sample, i.e., z_1), and $\hat{\sigma}_t^2$ is the Bayesian posterior estimation error ($\hat{\sigma}_t^2$ is initialized as the reference sensing accuracy given by the temperature sensor datasheet); 2) if the periodic cycle ends, then the smart sensor node considers the last Bayesian estimate as the truth measurement and submit it and the corresponding Bayesian estimation error to the Web server according to our proposed algorithm or the algorithms compared against ours; and 3) the server picks the Bayesian estimate with minimum Bayesian estimation error from all the estimates submitted and sees it as MEE

$$\hat{\mu}_t = \frac{\sigma_{z,t}^2 \hat{\mu}_{t-1} + z_t \hat{\sigma}_{t-1}^2}{\sigma_{z,t}^2 + \hat{\sigma}_{t-1}^2}, \quad \hat{\sigma}_t^2 = \frac{\sigma_{z,t}^2 \hat{\sigma}_{t-1}^2}{\sigma_{z,t}^2 + \hat{\sigma}_{t-1}^2}. \quad (20)$$

In the simulations, the software package of MATLAB Simulink Communication Block Set in [39] was used to establish the simulation environment: 1) the smart sensor nodes and Web server were both simulated by existing sensor models in Simulink; 2) the network channel was modeled by additive white Gaussian noise (AWGN) channel in Simulink allowing the setting of channel capability limit (according with Fig. 4, we replaced Bluetooth in [39] with Wi-Fi/4G network); and 3) the message collision and loss were calculated by path loss block in Simulink. To emulate the real scenario, simulations share the same noise with field experiments, i.e., we added the white Gaussian noise with mean value 0 and variance 0.5^2 to the simulated observations. Its spectrum is a uniform distribution with a magnitude of 0.5^2 .

B. Parameter Identification

Some parameters are identified before simulations and field experiments. We briefly state how to complete this. By (16) or (17) or Algorithm 1, the parameters should be identified are c and ε . For c , an estimate-submission package from the smart sensor node contains 800 B, and a Web service response package contains 200 B, without data block on the Internet, so $c = 800 + 200 = 1000$ B. For ε , it is calculated by (21), where C_{with} and C_{without} are the accumulative network traffic consumption, with and without data block, respectively, and n_ε

⁹See <https://wenku.baidu.com/view/8958c21314791711cc79174a.html>.

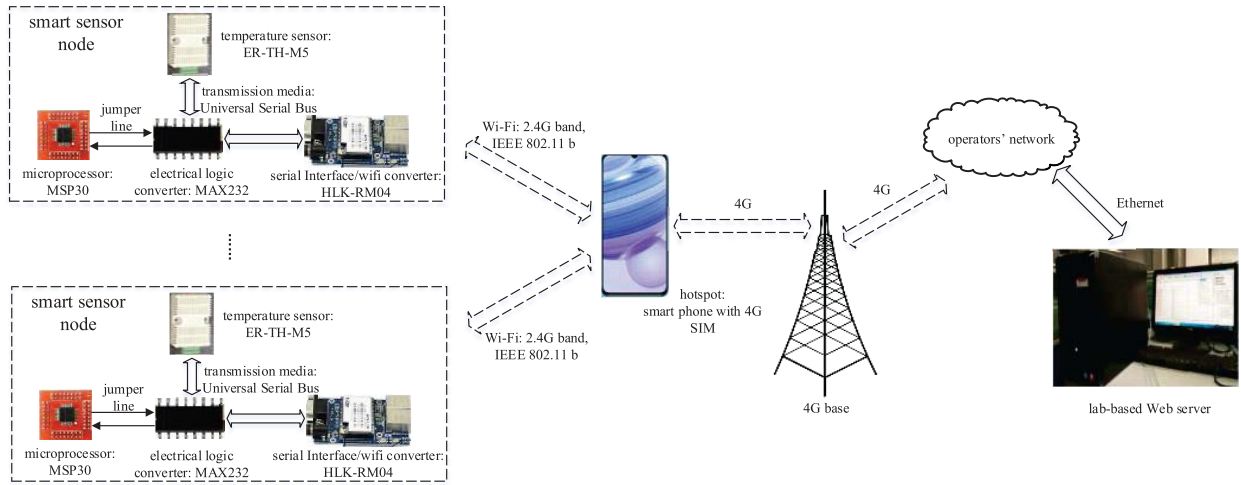


Fig. 4. Architecture of the experimental system. The smart sensor nodes work in the remote field area, while the Web server runs in the laboratory. There exists a 4G instead of the Wi-Fi network in the field area, while the smart sensor nodes have Wi-Fi instead of 4G function. Because of the higher cost of 4G than Wi-Fi and the limit of the experimental budget, it is infeasible to equip each smart sensor node with a 4G module and a subscriber identity module (SIM) card. The dashed two-way arrows denote wireless communication channels.

is the number of ε introduced by data blocks that are caused by sensor nodes submitting estimates in a shared Wi-Fi network

$$\varepsilon = \frac{C_{\text{with}} - C_{\text{without}}}{n_{\varepsilon}} \quad (21)$$

We let ten smart sensor nodes submit their estimates successively. This means that $C_{\text{without}} = 1000 \times 10 = 10000$ B and $n_{\varepsilon} = 55$ (see footnote 10).¹⁰ In the test, we measured $C_{\text{with}} = 10055$ B, so we have $\varepsilon \approx (10055 - 10000)/55 = 1$ B. To emulate the real scenario, simulations and field experiments share the same c and ε and are both carried out by the following three parts.¹¹

- 1) Carry out experiments under ten pairs of (n, Q) , as shown in Table I.
- 2) Carry out experiments under ten pairs of (n, Q) , as shown in Table II, according to changing the number n of sensor nodes at the same level of the background of network traffic Q .
- 3) Carry out experiments under ten pairs of (n, Q) , as shown in Table III, according to changing the background of network traffic Q while keeping the number n of sensor nodes unchanged.

C. Simulation Results

Thirty replications of the simulation are performed in total, the same as the field experiments. The 30 simulation results consist of three parts and are concluded, respectively, as the

¹⁰The hotspot (smartphone) always consumes network traffic even though there is no estimate-submission due to the reason that some built-in applications cannot be completely shut down and always lead to network traffic cost. The network data corresponding to such consumption and the estimate-submissions jointly produce data blocks. More estimate-submissions make more data blocks, and the network traffic consumed by an estimate-submission varies under a different number of blocks. ε denotes the average value of such variation per additional estimate submitted to the Web server. Therefore, for ten sensor nodes successively submitting estimates, the numbers of ε introduced by the first, second, ..., tenth estimate-submissions are 1, 2, ..., 10, respectively, summing to 55. ε may be, of course, positive, 0, or negative.

¹¹Because of the limits of the experiment field, the field experiments do not share the same number of sensor nodes with simulations.

TABLE I
TEN PAIRS OF (n, Q) IN THE SIMULATION AND FIELD EXPERIMENT
(n IS THE NUMBER OF ALL SENSOR NODES, AND Q IS THE AVAILABLE NETWORK TRAFFIC)

| (n, Q) in the simulation | (n, Q) in the field experiment |
|----------------------------|----------------------------------|
| (200, 150.0 KB) | (20, 13.70 KB) |
| (210, 158.3 KB) | (21, 14.39 KB) |
| (220, 150.7 KB) | (22, 15.07 KB) |
| (230, 165.8 KB) | (23, 15.76 KB) |
| (240, 180.8 KB) | (24, 16.44 KB) |
| (250, 188.4 KB) | (25, 17.13 KB) |
| (260, 195.9 KB) | (26, 17.81 KB) |
| (270, 203.5 KB) | (27, 18.50 KB) |
| (280, 211.0 KB) | (28, 19.18 KB) |
| (290, 218.6 KB) | (29, 19.87 KB) |

TABLE II
TEN PAIRS OF (n, Q) IN THE SIMULATION AND FIELD EXPERIMENT
ACCORDING TO CHANGING THE NUMBER n OF SENSOR NODES AT THE SAME LEVEL OF THE BACKGROUND OF NETWORK TRAFFIC Q , WHERE 284.138, AND 16.785 kB ARE THE AVERAGE VALUE OF Q IN THE FIRST AND SECOND COLUMNS, RESPECTIVELY, OF TABLE I

| (n, Q) in the simulation | (n, Q) in the field experiment |
|----------------------------|----------------------------------|
| (200, 184.7 KB) | (20, 16.79 KB) |
| (210, 184.7 KB) | (21, 16.79 KB) |
| (220, 184.7 KB) | (22, 16.79 KB) |
| (230, 184.7 KB) | (23, 16.79 KB) |
| (240, 184.7 KB) | (24, 16.79 KB) |
| (250, 184.7 KB) | (25, 16.79 KB) |
| (260, 184.7 KB) | (26, 16.79 KB) |
| (270, 184.7 KB) | (27, 16.79 KB) |
| (280, 184.7 KB) | (28, 16.79 KB) |
| (290, 184.7 KB) | (29, 16.79 KB) |

form of standard box plot in Fig. 5(a)–(c), which displays the probability of the Web server obtaining the MEE under ten pairs of (n, Q) set, as shown in the first column of Tables I–III, respectively. From Fig. 5(a)–(c), we can find that NTPTA does a better job than PDTA, NTEA, and WSRA in maximizing the probability of the Web server obtaining the MEE no matter whether (n, Q) comes from Tables I, II or III. Especially, the average probabilities of the 30 simulations by NTPTA, PDTA, NTEA, and WSRA are 0.9143, 0.8673, 0.8613, and 0.7296, respectively, which shows that NTPTA leads to higher probability than PDTA, NTEA, and WSRA with 5.42%, 6.15%, and 25.31%, respectively, and validates

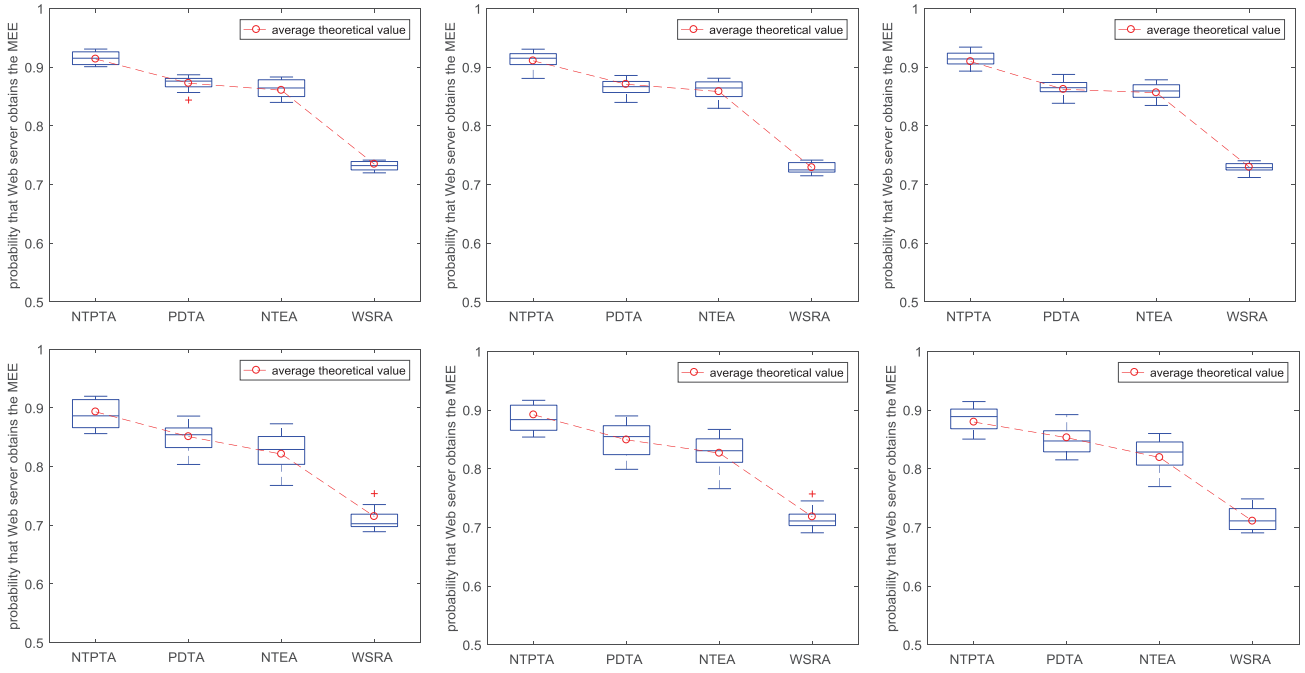


Fig. 5. (a)–(c) Ten simulation results under ten pairs of (n, Q) as shown in the first column of Tables I–III, respectively [(b) concludes the simulation results according to changing the number n of sensor nodes at the same level of the background of network traffic Q , and (c) concludes the simulation results according to changing the background of network traffic Q while keeping the number n of sensor nodes unchanged]. and (d)–(f) Ten field experiment results under ten pairs of (n, Q) as shown in the second column of Tables I–III, respectively [(e) concludes the failed experiment results according to changing the number n of sensor nodes at the same level of the background of network traffic Q , and (f) concludes the field experiment results according to changing the background of network traffic Q while keeping the number n of sensor nodes unchanged]. The dashed lines show the average theoretical probabilities.

TABLE III
TEN PAIRS OF (n, Q) IN THE SIMULATION AND FIELD EXPERIMENT ACCORDING TO CHANGING THE BACKGROUND OF NETWORK TRAFFIC Q WHILE KEEPING THE NUMBER n OF SENSOR NODES UNCHANGED, WHERE 110 AND 25 ARE THE APPROXIMATE AVERAGE VALUE OF n IN THE FIRST AND SECOND COLUMNS, RESPECTIVELY, OF TABLE I

| (n, Q) in the simulation | (n, Q) in the field experiment |
|----------------------------|----------------------------------|
| (250, 150.0 KB) | (25, 13.70 KB) |
| (250, 158.3 KB) | (25, 14.39 KB) |
| (250, 150.7 KB) | (25, 15.07 KB) |
| (250, 165.8 KB) | (25, 15.76 KB) |
| (250, 180.8 KB) | (25, 16.44 KB) |
| (250, 188.4 KB) | (25, 17.13 KB) |
| (250, 195.9 KB) | (25, 17.81 KB) |
| (250, 203.5 KB) | (25, 18.50 KB) |
| (250, 211.0 KB) | (25, 19.18 KB) |
| (250, 218.6 KB) | (25, 19.87 KB) |

the potential of NTPTA in improving the sensing accuracy for Web service-based data-collection systems.¹²

D. Field Experiment Results

Similar to the abovementioned simulation results, the 30 field experiment results consist of three parts

¹²Fig. 5 concludes the simulation and experiment results as the form of boxplots [40], where the bottom and top of the box are the first and third percentile, respectively, the bar inside the box is the median, the lower bar below the box is the lowest datum still within 1.5 interquartile range (IQR) of the first quartile (IQR is the difference between the third and first percentiles), the upper bar over the box is the highest datum still within 1.5 IQR of the third quartile, the lower red + below the box is the data outside of 1.5 IQR of the first quartile, the upper red + over the box is the data outside of 1.5 IQR of the third quartile (the lower and upper red + are both called extreme outliers), and the rest of lines are used to shape the box plot [33]. This figure shows the probability of the Web server guaranteeing the MEE under different pairs of channel capability limits and smart sensor nodes, not the error. Since the probability lies in $[0, 1]$, the magnitude of its boxes or bars is 1.

and are concluded respectively as the form of standard box plot in Fig. 5(d)–(f), which displays the probability of the Web server obtaining the MEE under ten pairs of (n, Q) set as shown in the second column of Tables I–III, respectively. From Fig. 5(d)–(f), we can find that NTPTA still does the best job. Especially, the average probability of the 30 field experiments by NTPTA, PDTA, NTEA, and WSRA are 0.8867, 0.8487, 0.8233, and 0.7134, respectively, which shows that NTPTA leads to higher probability than PDTA, NTEA, and WSRA with 4.48%, 7.70%, and 24.29%, respectively, and validates the potential of NTPTA in improving the sensing accuracy for Web service-based data-collection systems again.

We depicted the average theoretical probabilities as the form of dashed lines in Fig. 5, from which we can find that the overall trend of relative ranking of probabilities in the simulations, the field experiments, and theory are the same, despite the field experiments having the largest variance, due to something unknown and difficult to capture in the experiment field, such as sensor failures, network disturbance and instability, and data package losses [41].

In summary, under the same network traffic, a higher probability of the Web server obtaining the MEE can be achieved by NTPTA than PDTA, NTEA, and WSRA. Meanwhile, we can find something reasonable in the following.

- 1) Fig. 5(a)–(c) is similar to each other, and Fig. 5(d)–(f) is also similar to each other. The reason is that the available network traffics 184.7 and 16.79 kB in Table II approximate the average value of Q in the first and second columns, respectively, of Table I, and the numbers 250 and 25 of sensor nodes in Table III approximate

the average value of n in the first and second columns, respectively, of Table I.

- 2) Fig. 5 shows that the conclusion of Theorem 2, i.e., under the same level of allowed network traffic, $P_{\pi^*} > P_{\text{PDTA}} \geq P_{\text{NTEA}} > P_{\text{WSRA}}$ holds.

VII. CONCLUSION

For the Web service-based data-collection system with multiple smart sensor nodes periodically sampling and estimating the same unknown physical parameter of interest, this article has focused on designing a criterion for its smart sensor nodes to decide whether or not to submit their estimates to the Web server in order to maximize the probability of the server obtaining the minimum error estimate while meeting the condition of the required level of network traffic. Within an optimization-theoretical framework, a new algorithm NTPTA has been proposed and validated by theoretical analysis, simulation, and field experiments. Since smart sensor nodes nowadays are the foundations of CPS and IoT from the hardware or software points of view, and the Web service is an important communication element in the IoT system and the CPS, the estimate-submission management of smart sensor nodes based on NTPTA helps IoT system and CPS to operate in a network traffic efficient way while maintaining acceptable levels of estimation error, especially under network resource constraints.

Given that, in some cases, smart sensor nodes can communicate with each other, so our future work will focus on designing similar criteria for such scenarios.

APPENDIX

Proof of Theorem 1: First, we prove the sufficiency that some smart sensor nodes must take Δ_1 (see condition 3), so, without loss of generality, we consider that $s_{k_1}, s_{k_2}, \dots, s_{k_r}$ are these nodes ($r, k_1, k_2, \dots, k_r \in \{1, 2, \dots, n\}$), that is,

$$P\{a_i = \Delta_1\} = 1, \quad i \in \{k_1, k_2, \dots, k_r\}. \quad (22)$$

Then, by (9), (22), and the definition of $U_i(a_i)$, we have

$$\begin{aligned} U_i(a_i) &= P\{\delta_i = 1 | a_i = \Delta_1\} P\{a_i = \Delta_1\} \\ &\quad + P\{\delta_i = 1 | a_i = \Delta_2\} P\{a_i = \Delta_2\} \\ &= \frac{1}{n} \quad \forall i \in \{k_1, k_2, \dots, k_r\} \end{aligned} \quad (23)$$

$\min(e_1, e_2, \dots, e_n) \in$ or $\notin \{e_{k_1}, e_{k_2}, \dots, e_{k_r}\}$. If the latter holds, then $\forall s_i \in \mathbb{N} - \{s_{k_1}, s_{k_2}, \dots, s_{k_r}\}$, it always has the intelligence to consider that $s_{k_1}, s_{k_2}, \dots, s_{k_r}, \dots, s_{k_{j-1}}$ have submitted their estimates ahead of it ($j \in \{r+1, r+2, \dots, n\}$, $k_1, k_2, \dots, k_{j-1} \in \{1, 2, \dots, n\}$), and if $\min(e_{k_1}, e_{k_2}, \dots, e_{k_{j-1}}) \in \{e_{k_1}, e_{k_2}, \dots, e_{k_r}\}$, then there are valid grounds for it to perform the estimate-submission since its estimation error may be smaller than those of $s_{k_1}, s_{k_2}, \dots, s_{k_{j-1}}$. With this in mind and the conditions 1), 2), and 4) in Section II-A, we have

$$a_i = \begin{cases} \Delta_1, & \min(e_{k_1}, e_{k_2}, \dots, e_{k_{j-1}}) \in \{e_{k_1}, e_{k_2}, \dots, e_{k_r}\}, \\ & i = k_j \in \{1, 2, \dots, n\} - \{k_1, k_2, \dots, k_r\} \\ \Delta_2, & \text{otherwise} \end{cases} \quad (24)$$

further by (7) and (24), and considering that $\{1, 2, \dots, n\} - \{k_1, k_2, \dots, k_r\}$ can be relabeled as $\{k_{r+1}, k_{r+2}, \dots, k_n\}$, and s_i and a_i can be relabeled as s_{k_j} and a_{k_j} , respectively, we have

$$\begin{aligned} P\{a_i = \Delta_1\} &= P\{\min(e_{k_1}, e_{k_2}, \dots, e_{k_{j-1}}) \in \{e_{k_1}, e_{k_2}, \dots, e_{k_r}\}\} \\ &= \sum_{x=1}^r P\{e_{k_x} = \min(e_{k_1}, e_{k_2}, \dots, e_{k_{j-1}})\} \\ &= \frac{r}{j-1} \quad \forall i = k_j \in \{k_{r+1}, k_{r+2}, \dots, k_n\}. \end{aligned} \quad (25)$$

Then, by (9) and (25) and the definition of $U_i(a_i)$, we have

$$\begin{aligned} U_i(a_i, \mathbf{a}_{-i}) &= P\{\delta_i = 1 | a_i = \Delta_1\} P\{a_i = \Delta_1\} \\ &\quad + P\{\delta_i = 1 | a_i = \Delta_2\} P\{a_i = \Delta_2\} \\ &= \frac{r}{n(j-1)} \quad \forall i = k_j \in \{k_{r+1}, k_{r+2}, \dots, k_n\}. \end{aligned} \quad (26)$$

The condition 5) in Section II-A shows that the subscripts of s_1, s_2, \dots, s_n are arbitrary, so, by (23) and (26), and relabeling $s_{k_1}, s_{k_2}, \dots, s_{k_n}$ and $a_{k_1}, a_{k_2}, \dots, a_{k_n}$ as s_1, s_2, \dots, s_n and a_1, a_2, \dots, a_n , respectively, $U_i(a_i)$ can be written as

$$U_i(a_i) = \begin{cases} \frac{1}{n}, & i \in \{1, 2, \dots, r\} \\ \frac{r}{n(i-1)}, & i \in \{r+1, r+2, \dots, n\}. \end{cases} \quad (27)$$

$s_{k_1}, s_{k_2}, \dots, s_{k_r}$ must submit their estimates, so, by the definition of ε and λ_i , we have

$$C_i(a_i) = c + \lambda_i \varepsilon \quad i \in \{k_1, k_2, \dots, k_r\} \quad (28)$$

and by (25), we have

$$\begin{aligned} C_i(a_i) &= cP\{a_i = \Delta_1\} + 0P\{a_i = \Delta_2\} \\ &= \frac{r(c + \lambda_i \varepsilon)}{j-1} \quad \forall i = k_j \in \{k_{r+1}, k_{r+2}, \dots, k_n\}. \end{aligned} \quad (29)$$

Similar to (27), we have

$$C_i(a_i) = \begin{cases} c + \lambda_i \varepsilon, & i \in \{1, 2, \dots, r\} \\ \frac{r(c + \lambda_i \varepsilon)}{i-1}, & i \in \{r+1, r+2, \dots, n\}. \end{cases} \quad (30)$$

In a periodic cycle, the earlier estimate-submission suffers less data-block than the later ones, so (30) can be given by

$$C_i(a_i) = \begin{cases} c + i\varepsilon, & i \in \{1, 2, \dots, r\} \\ \frac{r(c + i\varepsilon)}{i-1}, & i \in \{r+1, r+2, \dots, n\}. \end{cases} \quad (31)$$

Equations (27) and (31) show that $U_i(a_i)$ and $C_i(a_i)$ both depend on r . Thus, we denote $P(r)$ as the cumulative probability of the Web server obtaining the $\min(e_1, e_2, \dots, e_n)$ and $C(r)$ as the cumulative network traffic consumed, that is,

$$P(r) = P_\pi = \sum_{i=1}^n U_i(a_i) \quad (32)$$

$$C(r) = C_\pi = \sum_{i=1}^n C_i(a_i). \quad (33)$$

Then, by (27) and (32), we have

$$P(r) = \frac{r}{n} + \sum_{i=r+1}^n \left(\frac{1}{n}\right) \left(\frac{r}{i-1}\right) \quad (34)$$

and by (31) and (33), we have

$$C(r) = \sum_{i=1}^r (c + i\varepsilon) + \sum_{i=r+1}^n \frac{r(c + i\varepsilon)}{i-1}. \quad (35)$$

Therefore, (14) can be specifically rewritten as

$$\begin{aligned} & \max_{r \in \{1, 2, \dots, n\}} \left(\frac{r}{n} + \sum_{i=r+1}^n \left(\frac{1}{n}\right) \left(\frac{r}{i-1}\right) \right) \\ & \text{s.t. } \sum_{i=1}^r (c + i\varepsilon) + \sum_{i=r+1}^n \frac{r(c + i\varepsilon)}{i-1} \leq Q. \end{aligned} \quad (36)$$

By (34), we have

$$P(r+1) - P(r) = \sum_{i=r+2}^n \left(\frac{1}{n}\right) \left(\frac{1}{i-1}\right) \geq 0 \quad (37)$$

and by (35), we have

$$C(r+1) - C(r) = \varepsilon + \sum_{i=r+2}^n \frac{c + i\varepsilon}{i-1} > 0 \quad (38)$$

which means that $P(r)$ and $C(r)$ are both increasing functions of r . Thus, the solution denoted by r^* for (36) is given by

$$r^* = \max\{r \mid C(r) \leq Q, r \in \{1, 2, \dots, n\}\} \quad (39)$$

and

$$\max_{r \in \{1, 2, \dots, n\}} P(r) = P(r^*). \quad (40)$$

Now, by (32), (34), and (35), we have (16) and (18).

By (34), we have

$$P(r) = \frac{r}{n} + \frac{r}{n} \sum_{i=r+1}^n \left(\frac{1}{n}\right) \left(\frac{1}{i-1}\right) = \frac{r}{n} + \frac{r}{n} \sum_{y=\frac{r}{n}}^{\frac{n-1}{n}} \left(\frac{1}{n}\right) \left(\frac{1}{y}\right) \quad (41)$$

so, for large n , we have

$$P(r) \approx \frac{r}{n} + \frac{r}{n} \int_{\frac{r}{n}}^1 \left(\frac{1}{y}\right) dy = \frac{r}{n} - \frac{r}{n} \ln \frac{r}{n}. \quad (42)$$

By (35), we have

$$C(r) = r \left(c + n\varepsilon - \frac{(r-1)\varepsilon}{2} \right) + r(c + \varepsilon) \sum_{y=\frac{r}{n}}^{\frac{n-1}{n}} \left(\frac{1}{n}\right) \left(\frac{1}{y}\right) \quad (43)$$

so, for large n , we have

$$C(r) \approx r \left(c + n\varepsilon - \frac{(r-1)\varepsilon}{2} \right) - r(c + \varepsilon) \ln \frac{r}{n}. \quad (44)$$

Now, by (32), (39), (40), (42), and (44), we have (17) and (19).

Second, we prove the necessity that combining (4), (11), (14), (36), (39), and (40), we have

$$\begin{aligned} P_{\pi^*} &= \max_{\pi \in G} \sum_{i=1}^n \frac{P\{a_i = \Delta_1\}}{n} = \max_{r \in \{1, 2, \dots, n\}} P(r) \\ &= P(r^*) = \frac{r^*}{n} + \sum_{i=r^*+1}^n \left(\frac{1}{n}\right) \left(\frac{r^*}{i-1}\right) \end{aligned} \quad (45)$$

and by (15), we have

$$\begin{aligned} & \sum_{i=1}^{r^*} \frac{P\{a_i = \Delta_1\}}{n} + \sum_{i=r^*+1}^n \frac{P\{a_i = \Delta_1\}}{n} \\ &= \frac{r^*}{n} + \sum_{i=r^*+1}^n \left(\frac{1}{n}\right) \left(\frac{r^*}{i-1}\right) = \text{RHS}(45). \end{aligned} \quad (46)$$

By (30) and (31), we have

$$\begin{aligned} & \sum_{i=1}^{r^*} (c + \lambda_i \varepsilon) P\{a_i = \Delta_1\} + \sum_{i=r^*+1}^n (c + \lambda_i \varepsilon) P\{a_i = \Delta_1\} \\ &= \sum_{i=1}^{r^*} (c + i\varepsilon) P\{a_i = \Delta_1\} + \sum_{i=r^*+1}^n (c + i\varepsilon) P\{a_i = \Delta_1\} \end{aligned} \quad (47)$$

and combining (4), (13), (14), (36), (39), and (40), we have

$$\begin{aligned} Q &\geq \max_{\pi \in G} C_{\pi} = \max_{\pi \in G} \sum_{i=1}^n (c + \lambda_i \varepsilon) P\{a_i = \Delta_1\} \\ &= \max_{r \in \{1, 2, \dots, n\}} C(r) = C(r^*) \\ &= \sum_{i=1}^{r^*} (c + i\varepsilon) + \sum_{i=r^*+1}^n \frac{r^*(c + i\varepsilon)}{i-1} \end{aligned} \quad (48)$$

and then, by (15) and (47), we have

$$\begin{aligned} & \sum_{i=1}^{r^*} (c + i\varepsilon) P\{a_i = \Delta_1\} + \sum_{i=r^*+1}^n (c + i\varepsilon) P\{a_i = \Delta_1\} \\ &= \sum_{i=1}^{r^*} (c + i\varepsilon) + \sum_{i=r^*+1}^n \frac{r^*(c + i\varepsilon)}{i-1} = \text{RHS}(48) \end{aligned} \quad (49)$$

where r^* is determined by (16) or (17) (for large n). This then completes the proof of Theorem 1. \square

Proof of Theorem 2: Let C_{π^*} , C_{PDTA} , C_{NTEA} , and C_{WSRA} denote, respectively, the network traffic consumed by NTPTA, PDTA, NTEA, and WSRA. First, we compare C_{π^*} with C_{PDTA} under $P_{\pi^*} = P_{\text{PDTA}}$. By (18), we have

$$P_{\pi^*} = \sum_{i=1}^{r^*} \frac{1}{n} + \sum_{i=r^*+1}^n \left(\frac{1}{n}\right) \left(\frac{r^*}{i-1}\right) \quad (50)$$

and by (16) and (35), we have

$$C_{\pi^*} = \sum_{i=1}^{r^*} (c + i\varepsilon) + \sum_{i=r^*+1}^n \frac{r^*(c + i\varepsilon)}{i-1}. \quad (51)$$

Let Q_{PDTA} denote the equivalent network traffic consumed by PDTA without data block; then, we have

$$P_{\text{PDTA}} = \sum_{i=1}^{r^*} \left(\frac{1}{n}\right) \left(\frac{Q_{\text{PDTA}}}{nc}\right) + \sum_{i=r^*+1}^n \left(\frac{1}{n}\right) \left(\frac{Q_{\text{PDTA}}}{nc}\right) \quad (52)$$

$$C_{\text{PDTA}} = \sum_{i=1}^{r^*} (c+i\varepsilon) \frac{Q_{\text{PDTA}}}{nc} + \sum_{i=r^*+1}^n (c+i\varepsilon) \frac{Q_{\text{PDTA}}}{nc}. \quad (53)$$

By our running assumption, the level Q is not sufficient to support all the smart sensor nodes to submit their estimates, so we have

$$\sum_{i=1}^n (c+i\varepsilon) > Q \quad (54)$$

and further have

$$Q_{\text{PDTA}} \leq \frac{2Q}{n(2c+(n+1)\varepsilon)} nc < nc. \quad (55)$$

By $P_{\pi^*} = P_{\text{PDTA}}$, (50), and (52), we have

$$\sum_{i=1}^{r^*} \left(1 - \frac{Q_{\text{PDTA}}}{nc}\right) + \sum_{i=r^*+1}^n \left(\frac{r^*}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right) = 0 \quad (56)$$

and by (51) and (53), we have

$$\begin{aligned} C_{\pi^*} - C_{\text{PDTA}} &= \sum_{i=1}^{r^*} (c+i\varepsilon) \left(1 - \frac{Q_{\text{PDTA}}}{nc}\right) \\ &+ \sum_{i=r^*+1}^n (c+i\varepsilon) \left(\frac{r^*}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right). \end{aligned} \quad (57)$$

Further by (56), we have (note that c is a positive constant)

$$\begin{aligned} C_{\pi^*} - C_{\text{PDTA}} &= \sum_{i=1}^{r^*} i\varepsilon \left(1 - \frac{Q_{\text{PDTA}}}{nc}\right) + \sum_{i=r^*+1}^n i\varepsilon \left(\frac{r^*}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right). \end{aligned} \quad (58)$$

By (55) and (56), we have

$$\sum_{i=r^*+1}^n \left(\frac{r^*}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right) < 0. \quad (59)$$

By (58), we have

$$\begin{aligned} C_{\pi^*} - C_{\text{PDTA}} &\leq \sum_{i=r^*+1}^n (i-r^*)\varepsilon \left(\frac{r}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right) \\ &+ r^*\varepsilon \left(\sum_{i=1}^{r^*} \left(1 - \frac{Q_{\text{PDTA}}}{nc}\right) + \sum_{i=r^*+1}^n \left(\frac{r^*}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right)\right) \end{aligned} \quad (60)$$

and then, combining with (56) and (59), we have (note that ε is a positive constant)

$$\begin{aligned} C_{\pi^*} - C_{\text{PDTA}} &\leq \sum_{i=r^*+1}^n (i-r^*)\varepsilon \left(\frac{r}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right) \\ &\leq \varepsilon \sum_{i=r^*+1}^n \left(\frac{r}{i-1} - \frac{Q_{\text{PDTA}}}{nc}\right) < 0 \end{aligned} \quad (61)$$

which means that $C_{\pi^*} < C_{\text{PDTA}}$ under $P_{\pi^*} = P_{\text{PDTA}}$. Network traffic and sensing accuracy are two contradictory metrics, so, under $C_{\pi^*} = C_{\text{PDTA}}$, we have

$$P_{\pi^*} > P_{\text{PDTA}}. \quad (62)$$

Second, we compare P_{PDTA} with P_{NTEA} under $C_{\text{PDTA}} = C_{\text{NTEA}} = Q$. By Algorithms 2 and 3, we have

$$P_{\text{PDTA}} = \frac{2cQ}{nc(2c+(n+1)\varepsilon)} \quad (63)$$

$$P_{\text{NTEA}} = \frac{1}{n} \left\lfloor \frac{2cQ}{c(2c+(n+1)\varepsilon)} \right\rfloor \quad (64)$$

and hence we have

$$\begin{aligned} P_{\text{PDTA}} - P_{\text{NTEA}} &= \frac{1}{n} \left(\frac{2Q}{2c+(n+1)\varepsilon} - \left\lfloor \frac{2Q}{2c+(n+1)\varepsilon} \right\rfloor \right) \\ &\geq 0 \end{aligned} \quad (65)$$

which means that

$$P_{\text{PDTA}} \geq P_{\text{NTEA}}. \quad (66)$$

Third, we compare P_{NTEA} with P_{WSRA} under $C_{\text{NTEA}} = C_{\text{WSRA}} = Q$. By Algorithm 4, we have

$$P_{\text{WSRA}} = \frac{1}{n} \left\lfloor \frac{Q}{2c+(n+1)\varepsilon} \right\rfloor \quad (67)$$

and, hence, combining with (64), we have

$$P_{\text{NTEA}} > P_{\text{WSRA}}. \quad (68)$$

Finally, by (62), (66), and (67), we have

$$P_{\pi^*} > P_{\text{PDTA}} \geq P_{\text{NTEA}} > P_{\text{WSRA}} \quad (69)$$

and this concludes the proof of Theorem 2. \square

REFERENCES

- [1] T. Bandyszak, M. Daun, B. Tenbergen, P. Kuhs, S. Wolf, and T. Weyer, "Orthogonal uncertainty modeling in the engineering of cyber-physical systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1250–1265, Jul. 2020.
- [2] M. Ben-Daya, E. Hassini, and Z. Bahroun, "Internet of Things and supply chain management: A literature review," *Int. J. Prod. Res.*, vol. 57, nos. 15–16, pp. 4719–4742, Aug. 2019.
- [3] C. Srinivasan, B. Rajesh, P. Saikalyan, K. Premsagar, and E. S. Yadav, "A review on the different types of Internet of Things (IoT)," *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 1, pp. 154–158, 2019.
- [4] Z. Shelby and S. Ukkola, "Method, apparatus and system for Web service management," U.S. Patent 9614877, Apr. 4, 2017.
- [5] R. Frank, *Understanding Smart Sensors*. Norwood, MA, USA: Artech House, 2013.
- [6] A. L. Lemos, F. Daniel, and B. Benatallah, "Web service composition: A survey of techniques and tools," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 33–43, 2016.
- [7] C. Hou and Q. Zhao, "Optimization of Web service-based control system for balance between network traffic and delay," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1152–1162, Jul. 2018.
- [8] J. J. Livingston and A. Umamakeswari, "Internet of Things application using IP-enabled sensor node and Web server," *Indian J. Sci. Technol.*, vol. 8, no. S9, pp. 207–212, 2015.
- [9] L. A. Caceres Najarro, I. Song, and K. Kim, "Differential evolution with opposition and redirection for source localization using RSS measurements in wireless sensor networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1736–1747, Oct. 2020.
- [10] A. Prorok, "Robust assignment using redundant robots on transport networks with uncertain travel time," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 2025–2037, Oct. 2020.

- [11] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao, "Tiny Web services: Design and implementation of interoperable and evolvable sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 253–266.
- [12] F. C. Delicato, P. F. Pires, L. Pinnez, L. Fernando, and L. Da Costa, "A flexible Web service based architecture for wireless sensor networks," in *Proc. 23rd Int. Conf. Distrib. Comput. Syst.*, 2003, pp. 730–735.
- [13] A. Ouammi, Y. Achour, D. Zejli, and H. Dagdougui, "Supervisory model predictive control for optimal energy management of networked smart greenhouses integrated microgrid," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 117–128, Jan. 2020.
- [14] C. Buckl *et al.*, "Services to the field: An approach for resource constrained sensor/actor networks," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2009, pp. 476–481.
- [15] Y.-W. Hong, W.-J. Huang, F.-H. Chiu, and C.-C.-J. Kuo, "Cooperative communications in resource-constrained wireless networks," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 47–57, May 2007.
- [16] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proc. 1st ACM Conf. Wireless Netw.*, 2008, pp. 77–88.
- [17] M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi, "Middleware for wireless sensor networks: A survey," *J. Comput. Sci. Technol.*, vol. 23, no. 3, pp. 305–326, 2008.
- [18] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.
- [19] D. K. Barry, *Web Services and Service-Oriented Architectures*. Amsterdam, The Netherlands: Elsevier, 2003.
- [20] F. Hirsch, J. Kemp, and J. Ilkka, *Mobile Web Services: Architecture and Implementation*. Hoboken, NJ, USA: Wiley, 2007.
- [21] K. P. Birman, *Reliable Distributed Systems: Technologies, Web Services, and Applications*. New York, NY, USA: Springer, 2005.
- [22] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, May 2014.
- [23] X.-Y. Liu *et al.*, "CDC: Compressive data collection for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2188–2197, Aug. 2015.
- [24] R. Kyusakov, J. Eliasson, J. Delsing, J. van Deventer, and J. Gustafsson, "Integration of wireless sensor and actuator nodes with IT infrastructure using service-oriented architecture," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 43–51, Feb. 2013.
- [25] N. Glombitza, D. Pfisterer, and S. Fischer, "Using state machines for a model driven development of Web service-based sensor network applications," in *Proc. Workshop Softw. Eng. Sensor Netw. Appl. (ICSE)*, 2010, pp. 2–7.
- [26] S. N. Han, G. M. Lee, and N. Crespi, "Semantic context-aware service composition for building automation system," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 752–761, Feb. 2014.
- [27] A. Khan, A. Abdullah, M. Anisi, and J. Bangash, "A comprehensive study of data collection schemes using mobile sinks in wireless sensor networks," *Sensors*, vol. 14, no. 2, pp. 2510–2548, Feb. 2014.
- [28] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Hoboken, NJ, USA: Wiley, 2007.
- [29] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. Hoboken, NJ, USA: Wiley, 2009.
- [30] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [31] C. De Persis and S. Grammatico, "Distributed averaging integral Nash equilibrium seeking on networks," *Automatica*, vol. 110, pp. 108548–108558, 2019.
- [32] S. Yüksel and T. Başar, *Stochastic Networked Control Systems: Stabilization and Optimization under Information Constraints*. New York, NY, USA: Springer, 2013.
- [33] C. Hou and Q. Zhao, "Stopping-time management of smart sensing nodes based on tradeoffs between accuracy and power consumption," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2472–2485, Sep. 2017.
- [34] I. Mahgoub and M. Ilyas, *Sensor Network Protocols*. Boca Raton, FL, USA: CRC Press, 2016.
- [35] *MAX232 Datasheet*, MaximIntegrated, San Jose, CA, USA, 2017.
- [36] *HLK-RM04 Datasheet*, HLK-RM04, Shenzhen, Guangdong, China, 2017.
- [37] *MSP430F149 Datasheet*, Texas Instrum., Dallas, TX, USA, 2017.
- [38] *ER-TH-M5 Datasheet*, Easy Run, Beijing, China, 2017.
- [39] Q. I. Ali, "Simulation framework of wireless sensor network (WSN) using MATLAB/simulink software," *MATLAB-A Fundam. Tool Sci. Comput. Eng. Appl.*, vol. 2, pp. 264–284, Sep. 2012.
- [40] D. F. Williamson, R. A. Parker, and J. S. Kendrick, "The box plot: A simple visual method to interpret data," *Ann. Internal Med.*, vol. 110, no. 11, pp. 916–921, 1989.
- [41] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in Internet of Things: A state-of-the-art survey," *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, Sep. 2016.



Chen Hou received the B.E. degree in automation from the College of Information and Electrical Engineering, China Agricultural University, Beijing, China, in 2012, and the M.S. and Ph.D. degrees in control science and engineering from the Department of Automation, Tsinghua University, Beijing, in 2015 and 2019, respectively.

He was a Visiting Scholar with the College of Engineering, University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2018. His research interests include the operation and control of the Internet of Things (IoT) under resource constraints.



Qianchuan Zhao (Senior Member, IEEE) received the B.E. degree in automatic control and the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in control theory and its applications from Tsinghua University, Beijing, China, in 1992 and 1996, respectively.

He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, in 2000, and Harvard University, Cambridge, MA, USA, in 2002. He was a Visiting Professor with Cornell University, Ithaca, NY, USA, in 2006. He is currently

a Professor and the Director of the Center for Intelligent and Networked Systems, Department of Automation, Tsinghua University. He has authored or coauthored more than 80 research articles in peer-reviewed journals and conferences. His current research interests include the control and optimization of complex networked systems with applications in smart buildings, smart grid, and manufacturing automation.

Dr. Zhao received the 2009 China National Nature Science Award for the Project Optimization Theory and Optimization for Discrete Event Dynamic System. He serves as the Chair for the Technical Committee on Smart Buildings of the IEEE Robotics and Automation Society. He is an Associate Editor of the *Journal of Optimization Theory and Applications*, the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.



Tamer Başar (Life Fellow, IEEE) received the B.S.E.E. degree from Robert College, Istanbul, and the M.S., M.Phil., and Ph.D. degrees from Yale University.

He is with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, where he holds the academic positions of Swanlund Endowed Chair; Center for Advanced Study Professor of Electrical and Computer Engineering; Research Professor with the Coordinated Science Laboratory; and a Research Professor with the Information Trust Institute. He is

also the Director of the Center for Advanced Study. He has over 900 publications in systems, control, communications, networks, and dynamic games, including books on noncooperative dynamic game theory, robust control, network security, wireless and communication networks, and stochastic networked control. His current research interests include stochastic teams, games, and networks; distributed optimization and learning; security; energy systems; and cyber-physical systems.

Dr. Başar is a member of the U National Academy of Engineering, a member of the European Academy of Sciences, and a Fellow of the International Federation of Automatic Control (IFAC) and Society for Industrial and Applied Mathematics (SIAM). He has served as a President of the IEEE Control Systems Society (CSS), International Society of Dynamic Games (ISDG), and American Automatic Control Council (AACC). He has received several awards and recognitions over the years, including the highest awards of IEEE CSS, IFAC, AACC, and ISDG, the IEEE Control Systems Award, and a number of international honorary doctorates and professorships. He was the Editor-in-Chief of *Automatica* between 2004 and 2014, and is currently editor of several book series.