

Configuration Dissemination with State Feedback for Wireless Sensor Networking Systems

Heejung Byun¹ and Soo-Mi Yang^{2*}

¹Department of Information and Telecommunication Engineering, The University of Suwon,
Wauan-gil 17, Hwasungsi, Gyeonggi-do 18323, Korea

²Department of Information Engineering, The University of Suwon,
Wauan-gil 17, Hwasungsi, Gyeonggi-do 18323, Korea

(Received April 16, 2018; accepted March 20, 2019)

Keywords: wireless sensor networks, dissemination, monitoring of sensor state

A wireless sensor network (WSN) is a group of spatially dispersed sensors for monitoring the physical conditions of the environment and organizing the collected data at a central location. Since WSNs typically have no infrastructure, one of the challenges in WSNs is the lack of an appropriate method for an operator to interact with a WSN. Practical tasks for WSNs require sensor nodes adaptive to changes in WSN configuration, user input, and commands; hence, monitoring and interaction with a WSN are required. In this paper, a new scheme for configuration dissemination with the monitoring of the sensor state is proposed to control dynamically the operation of sensor nodes in real time. Moreover, a method is provided for the operator to observe the state of a sensor node, which helps the operator provide an appropriate control input to the WSN. We prove the stability properties of the proposed algorithm and show the simulation results.

1. Introduction

Wireless sensor networks (WSNs) consist of sink nodes and a large number of sensor nodes that coordinate autonomously. These networks are used to monitor physical or environmental conditions such as sound, pressure, and temperature, and propagate data through a network to a main location. However, sensor nodes are not yet capable of autonomously performing various tasks, and many tasks require coordination between an operator and a WSN. Therefore, the operator should recognize the state of the WSN and convey parameter adjustments or configurations to the WSN.

One of the prominent works regarding the configuration dissemination model involves the use of sensor protocols for information via negotiation (SPIN).⁽¹⁾ The authors focused on the efficient dissemination of individual sensor observations to all the sensors and proposed data descriptors to eliminate the chance of redundant transmission in WSNs. Other authors proposed a time division multiple access (TDMA)-based data dissemination protocol for WSNs

*Corresponding author: e-mail: smyang@suwon.ac.kr
<https://doi.org/10.18494/SAM.2019.2275>

to offer a degree of reliability.⁽²⁾ The Firecracker protocol⁽³⁾ used a combination of routing and broadcast principles to rapidly propagate data throughout WSNs. Trickle⁽⁴⁾ was proposed for propagating and maintaining code updates as fast as possible from the operator to all nodes in the network. This protocol considered the suppression and dynamic adjustment of the broadcast rate to limit transmissions among neighboring nodes. It only provided a mechanism by which a node might decide when to propagate the code. Deluge⁽⁵⁾ added a feature that supports the transfer of large data objects based on Trickle principles. Still other authors proposed a gossip-based approach where each node decides to forward a message to another node based on some probability.⁽⁶⁾ The problem with this gossip approach is that, if a source has very few neighbors, then the nodes will not gossip and the information dies out. Middle-ware has been proposed⁽⁷⁾ to provide controlled propagation to tune the process according to the desired reliability. In another case,⁽⁸⁾ a multirumor overwriting model was proposed. The model described the propagation and interaction of periodically sensed data, which were distributed by gossiping, without increasing the traffic on the network. Walker and coworkers^(9,10) focused on two methods of data propagation (flooding and consensus methods) and compared the ability of operators to manage the system to reach the desired goals. In the flooding method, each node explicitly matches the value of operator command, and in the consensus method, each node matches the average value of operator command of all the neighbors it senses. The data fusion support system⁽¹¹⁾ provides intelligent capabilities for real-time control access, so that the configurable integrated web-based support services are structured in the context of an ad-hoc wireless sensor network.

Existing works have focused on conveying the data between the nodes, but the WSN requires an operator to be able to observe the WSN states in response to operator commands and the effects of control input. This helps the user provide the next configuration. However, the monitoring of the sensor node state is particularly difficult for the operator, and limited research has probed into determining it. In this paper, an autonomous real-time algorithm for the configuration and monitoring of the WSN is proposed. The proposed method properly disseminates the WSN configuration by indirectly setting parameters within the framework of local rules in a decentralized manner. The proposed algorithm also allows the user to observe effectively the state of the sensor node by understanding WSN dynamics.

2. Proposed Algorithm

2.1 Model

We consider a WSN system consisting of $(N - 1)$ sensor nodes, a sink node (N -th node), and an operator. Let $\mathcal{N} = \{1, 2, \dots, N\}$ denote the set of sensor nodes in the WSN. The set of neighboring nodes of sensor node i ($1 \leq i \leq N$) is denoted as N_i and its cardinality is $|N_i|$. We propose a control model of the WSN system, where for the forward control loop, the operator controls the WSN by directly controlling the sink node, which affects dissemination to the rest of the sensor nodes based on the control law of each sensor node. For the backward control loop, the operator observes the state of the sensor nodes through the sink node that provides

feedback regarding the node state. No control can be exerted over individual sensor nodes, only over the WSN as a whole. To this end, we define the state of the WSN system as $\mathbf{x}(t)$ and denote $\mathbf{u}(t)$ as the desired configuration set that the operator wants to achieve with the WSN system at time t . Specifically, $\mathbf{x}(t)$ represents the vector of the states of all sensor nodes at time t , which can be indicated as vector $\mathbf{x}(t) = [x_1(t) \ x_2(t) \ \dots \ x_N(t)]^T$, where $\mathbf{x}_i(t) = [x_i^1(t) \ x_i^2(t) \ \dots \ x_i^n(t)]^T$ is the state vector of node i at time t and an individual node has n -dimensional state space. The state dynamics of the WSN can be defined as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)). \quad (1)$$

The operator interacts with the WSN by applying the control input to the sink node, while the other nodes interact with each other following the local control law of each node in the WSN:

$$\dot{\mathbf{x}}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)), \forall i, i \neq N. \quad (2)$$

The operator gives commands or informs the sink node of parameter change with $\mathbf{u}(t)$, after which information propagates according to the underlying control law of sensor node members. Let \mathbf{x}_N be the state of a sink node and the user input is directly applied to the sink node by letting $\mathbf{x}_N(t) = \mathbf{u}(t)$. Then, the dynamic for the sink node is

$$\dot{\mathbf{x}}_N(t) = 0. \quad (3)$$

According to Eqs. (2) and (3), the operator directly controls the state of the single sink node and the interactions between the sensor nodes are handled autonomously. That is, the operator's intended command is directly conveyed to the sink node and each sensor node averages the states of all neighbors it can sense, and then sets its state vector based on that average state. In this manner, the operator takes actions independent of the number of nodes and supervises the WSN as a single entity, resulting in a control complexity of $O(1)$.

2.2 Stability analysis

In this section, we will analyze the feasibility of the proposed algorithm for an operator to achieve desired WSN configurations. We denote a steady state \mathbf{x}_s , which is the state where the proposed system under the local control law can asymptotically converge. Let us denote $\mathbf{x}_s = [x_{1s} \ x_{2s} \ \dots \ x_{Ns}]^T$ as a steady state vector $\mathbf{x}(t)$, where \mathbf{x}_{is} is the steady state of sensor node i . We consider that the desired WSN configuration set $\mathbf{u}(t)$ is piecewise constant such that $\mathbf{u}(t) \approx \mathbf{u}$.

From Eq. (2), \mathbf{x}_{is} is expressed as

$$\mathbf{x}_{is} = \frac{1}{N_i} \left(1_{(i,1)} x_{1s} + 1_{(i,2)} x_{2s} + \dots + 1_{(i,N)} x_{Ns} \right), \forall i \in N, i \neq N, \quad (4)$$

where $\mathbf{1}_{(i,j)}$ is the indication function: $\mathbf{1}_{(i,j)} = 1$ indicating $j \in N_i$, otherwise, $\mathbf{1}_{(i,j)} = 0$, therefore

$\sum_{j \in N_i} \mathbf{1}_{(i,j)} = N_i$. Let $\tilde{\mathbf{x}}_s = [x_{1s} \ x_{2s} \ \dots \ x_{(N-1)s}]^T$. Then, Eq. (4) is rewritten as

$$\mathbf{A}\tilde{\mathbf{x}}_s = \mathbf{b}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -\frac{1}{N_1}\mathbf{1}_{(1,2)} & \dots & -\frac{1}{N_1}\mathbf{1}_{(1,N-1)} \\ -\frac{1}{N_2}\mathbf{1}_{(2,1)} & 1 & \vdots & -\frac{1}{N_2}\mathbf{1}_{(2,N-1)} \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{1}{N_{N-1}}\mathbf{1}_{(N-1,1)} & -\frac{1}{N_{N-1}}\mathbf{1}_{(N-1,2)} & \dots & -\frac{1}{N_{N-1}}\mathbf{1}_{(N-1,N-1)} \end{bmatrix},$$

$$\mathbf{b} = \left[\frac{1}{N_1}\mathbf{1}_{(1,N)}u \ \frac{1}{N_2}\mathbf{1}_{(2,N)}u \ \dots \ \frac{1}{N_{N-1}}\mathbf{1}_{(N-1,N)}u \right]^T.$$

Let \mathbf{L} be a Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where \mathbf{D} is the diagonal matrix and

$$A_{ij} = \begin{cases} 1 & \text{if } \mathbf{1}_{(i,j)} = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

Therefore, the elements of \mathbf{L} are given by

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{N_i} & \text{if } \mathbf{1}_{(i,j)} = 1. \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Consider a vector $\mathbf{v} = [x_1, x_2, \dots, x_N]^T$, then we will have the action of \mathbf{L} as

$$[\mathbf{L}\mathbf{v}]_i = x_i - \frac{1}{N_i} \sum_{j \in N_i, j \neq i} x_j. \tag{8}$$

We observe that $\mathbf{1} = [1, \dots, 1]^T$ is an eigenvector of \mathbf{L} with eigenvalue 0, since x_i always equals the average of its neighbors' values for this vector.

According to Eq. (5), \mathbf{A} has a Laplacian matrix such as \mathbf{L} and the constant \mathbf{x}_s are eigenvectors

of eigenvalue 0. In the proposed algorithm, the user input is directly applied to the N -th agent, and then we divide the following two cases: $\mathbf{1}_{(i,j)} = 1$ and $\mathbf{1}_{(i,j)} = 0$. In the case of $\mathbf{1}_{(i,j)} = 1$, obviously $\mathbf{x}_{Ns} = \mathbf{u}$. From Eq. (4), we obtain

$$N_i \mathbf{x}_{is} = (N_i - 1) \mathbf{x}_{is} + \mathbf{u}. \quad (9)$$

It shows that $\mathbf{x}_{is} = \mathbf{u}$, for all i . That is,

$$\mathbf{x}_s = [u \ u \ \dots \ u]^T. \quad (10)$$

In the case of $\mathbf{1}_{(i,j)} = 0$,

$$\sum_{j \in N_i, j \neq N} \mathbf{x}_{js} = N_i \mathbf{x}_{is}. \quad (11)$$

If there exists any $j \in N_i$ with $N \in N_j$, we obtain $\mathbf{x}_{is} = \mathbf{x}_{js}$. Finally, we obtain the following conclusion:

$$\begin{aligned} \mathbf{x}_s &= [x_{1s} \ x_{2s} \ \dots \ x_{Ns}]^T \\ &= [u \ u \ \dots \ u]^T. \end{aligned} \quad (12)$$

Therefore, we conclude that the operator can achieve some configuration described by $u(t)$ with the proposed system.

Next, we need to prove that the proposed system ensures stability. If the system is asymptotically stable, then the trajectory will converge to the steady state, described in Eqs. (10) and (12), as time goes to infinity. To derive the system stability, we define the following change of coordinates:

$$\mathbf{e}_i(t) = \mathbf{x}_i(t) - \mathbf{x}_{is}. \quad (13)$$

Then, we obtain the following representation of the system

$$\begin{aligned} \dot{\mathbf{e}}_i(t) &= \sum_{j \in N_i} (\mathbf{e}_j(t) - \mathbf{e}_i(t)) \\ &= -N_i \mathbf{e}_i(t) + \sum_{j \in N_i} (\mathbf{e}_j(t)). \end{aligned} \quad (14)$$

Defining $\mathbf{e}(t) = [e_1(t) \ e_2(t) \ \dots \ e_{N-1}(t)]^T$, our system can be represented in a matrix form as

$$\dot{\mathbf{e}}(t) = \mathbf{A}_s \mathbf{e}(t), \quad (15)$$

where

$$A_s = \begin{bmatrix} -N_1 & 1_{(1,2)} & \cdots & 1_{(1,N-1)} \\ 1_{(2,1)} & -N_2 & \cdots & 1_{(2,N-1)} \\ \vdots & \ddots & \ddots & \vdots \\ 1_{(N-1,1)} & 1_{(N-1,2)} & \cdots & -N_{N-1} \end{bmatrix}.$$

We use the Lyapunov theory to claim that the system defined by Eq. (15) will be convergent. The stability of solutions near a point of equilibrium is discussed by the Lyapunov theory. In simple terms, if the solutions that start out near an equilibrium point \mathbf{x}_e remain near \mathbf{x}_e forever, then \mathbf{x}_e is Lyapunov stable. More strongly, if \mathbf{x}_e is Lyapunov stable and all solutions that start out near \mathbf{x}_e converge to \mathbf{x}_e , then \mathbf{x}_e is asymptotically stable. From this theory, we can find a Lyapunov function $V(\mathbf{e})$ that satisfies the following conditions:

- $V(\mathbf{e})$ is positive definite;
- $\dot{V}(\mathbf{e})$ is negative definite.

We assume that there exist positive definite matrices \mathbf{P} and \mathbf{Q} , $\mathbf{P} = \mathbf{P}^T$, such that

$$\mathbf{P}A_s + A_s^T\mathbf{P} = -\mathbf{Q}. \quad (16)$$

Consider the Lyapunov function in the form

$$V(\mathbf{e}) = \frac{1}{2} \mathbf{e}^T \mathbf{P} \mathbf{e}. \quad (17)$$

Then, obviously $V(\mathbf{e}) \geq 0$ and we can obtain the time derivative $\dot{V}(\mathbf{e})$ as

$$\begin{aligned} \dot{V}(\mathbf{e}) &= \frac{1}{2} (\dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}}) \\ &= \frac{1}{2} \mathbf{e}^T (\mathbf{A}_s^T \mathbf{P} + \mathbf{P} \mathbf{A}_s) \mathbf{e} \\ &= -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} \leq 0. \end{aligned}$$

On the basis of the above analysis, both conditions of the Lyapunov function $V(\mathbf{e})$ have been satisfied; thus, we can consider that the system is stable and the error dynamics will converge to zero exponentially. This result guarantees the autonomous adaptation of the sensor node state according to the operator-defined WSN configuration, resulting in the asynchronous convergence of the WSN to the desired global behavior. For state monitoring, the sink node determines the difference between the states of its neighbors and the control input. The sink node feedbacks the difference to the operator. The closer the difference is to 0, the nearer the configuration of the entire network to completion. Therefore, the following command can be issued to the gateway node. According to the above stability analysis, we know that the states of all sensor nodes converge to the control input in the steady state. Therefore, the operator can

learn to understand node states and estimate the timing to give the next configuration inputs, given the feedback from the sink node only.

3. Simulation Results

3.1 Configuration

To evaluate the performance of our proposed system, we develop a simulation environment using MATLAB. The simulation area is $100 \times 100 \text{ m}^2$, where the entire network is divided into equally shaped grids, and the sensor nodes are uniformly deployed. We set $N = 50$, and the sensor nodes are arbitrarily connected. The sink node is denoted as R_{50} , which is chosen randomly by the operator. The channel capacity is set to 200 kbps, and the transmission and carrier sense ranges are set to 20 and 40 m, respectively.

3.2 Results

We compared the performance of the proposed method with that of the consensus method.⁽⁹⁾ The consensus algorithm is an asynchronous distributed protocol for distributed averaging, which aims to compute the average $\bar{x} = \sum_{i=1}^N \frac{x_i}{N}$. The consensus algorithm is used to achieve agreement on a single state among the distributed nodes and designed to achieve reliability in a network with multiple unreliable nodes. Each member in a system averages the values of all of its neighbors and adapts its own value toward that average.

Figures 1 and 2 illustrate the time trajectories of node states when the command input remains constant during runtime. The input is given by the operator, such as $u = 4$. Simulations are conducted over the interval from 0 to 5 s in 100 ms increments. Randomly chosen are 3 out of 49 nodes, namely, R_1 , R_2 , and R_3 . To show the performance of controlling the state

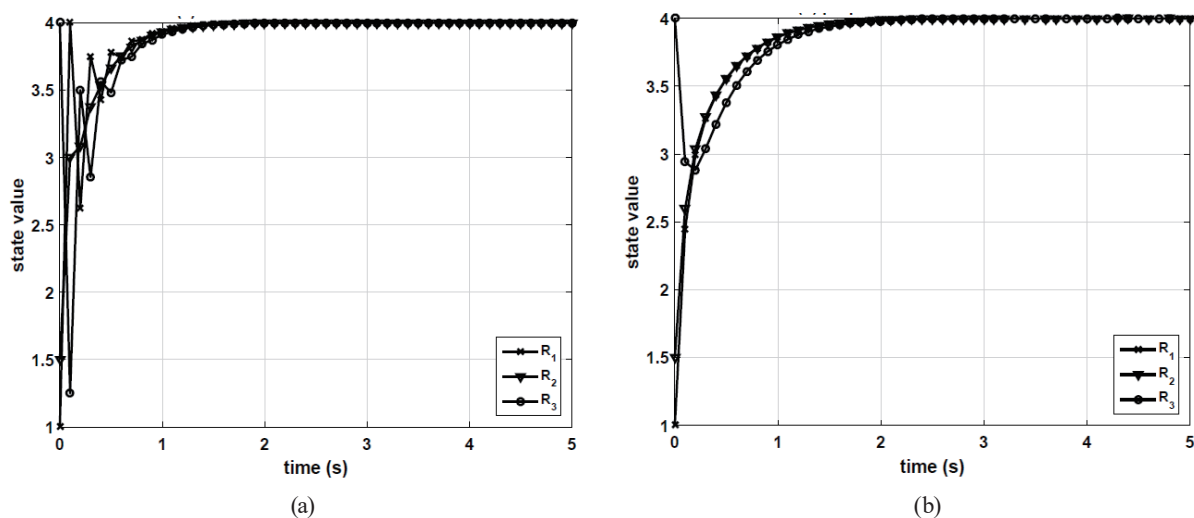


Fig. 1. Time trajectories of R_1 , R_2 , and R_3 without error: (a) consensus and (b) proposed methods.

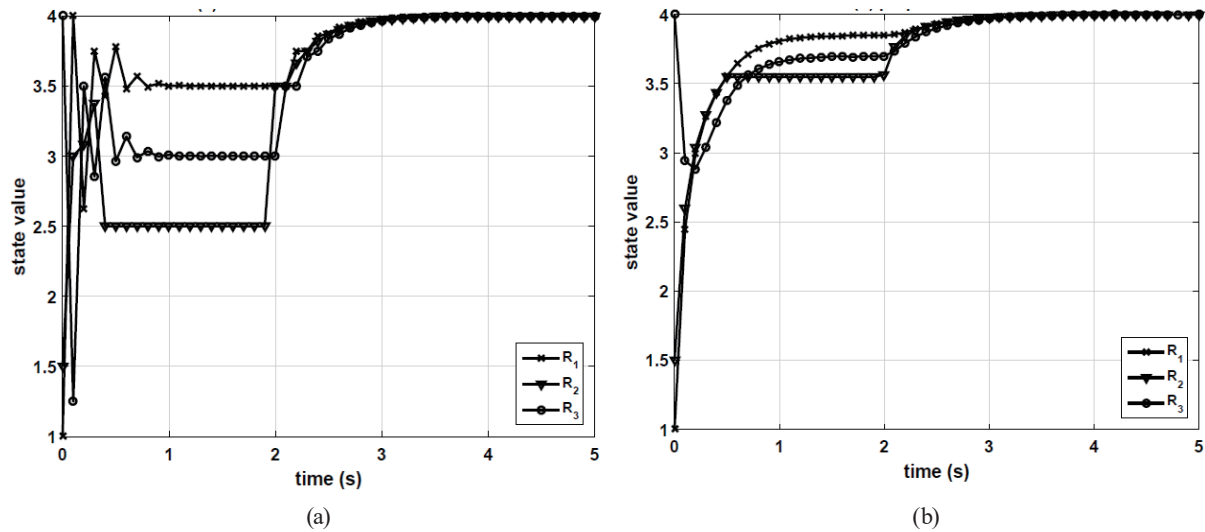


Fig. 2. Time trajectories of R_1 , R_2 , and R_3 with error: (a) consensus and (b) proposed methods.

of each node, we set the initial state for each node differently. Figures 1(a) and 1(b) illustrate the state changes with the consensus and proposed methods in the absence of error. Both the consensus and proposed methods converge the states of all sensor nodes to the desired state u despite the difference in the initial state of each node. As for the time taken to converge to the configuration by the operator, the consensus method is slightly faster, but both methods show similar performance characteristics for the error-free condition.

Figures 2(a) and 2(b) illustrate the state changes of the nodes under an error condition. The error condition is implemented so that the state value of R_2 is not updated between 0.5 and 2 s owing to the malfunction of R_2 . During [0.5, 2] s, R_2 exchanges the last updated state value with that of its neighbor sensor nodes. Both methods show that the state values of all nodes converge to the desired state after 3 s. However, for the consensus method, the malfunction of R_2 directly affects the state values of the neighboring nodes during [0.5, 2] s.

That is, in the consensus method, the state value of each node is sensitive to changes in its neighbor nodes' states; thus, a large difference in state value is observed between the nodes. On the other hand, the proposed method shows that the wrong behavior of R_2 does not have a significant effect on the state control of other nodes. As can be seen from Eq. (2), each node indirectly uses the state of the neighboring nodes in adjusting its state value, which makes it less susceptible to error conditions and have more robust performance.

Next, we examine the trajectories of states corresponding to control input changes during runtime. Simulations are conducted over the interval from 0 to 20 s in 100 ms increments. The configuration input is initially set to $u = 2$ and changed to $u = 4$ at 10 s. The error condition is implemented so that the state value of R_2 is not updated during [0.5, 2] s and [10.5, 12] s. Figures 3(a) and 3(b) respectively show the state adaptation behavior for the consensus and proposed methods according to control input changes. As shown in Fig. 3(a), the consensus method shows that the state values of R_1 and R_3 are largely affected by the state value of R_2 during the malfunction period, and the difference in state value between the nodes remains

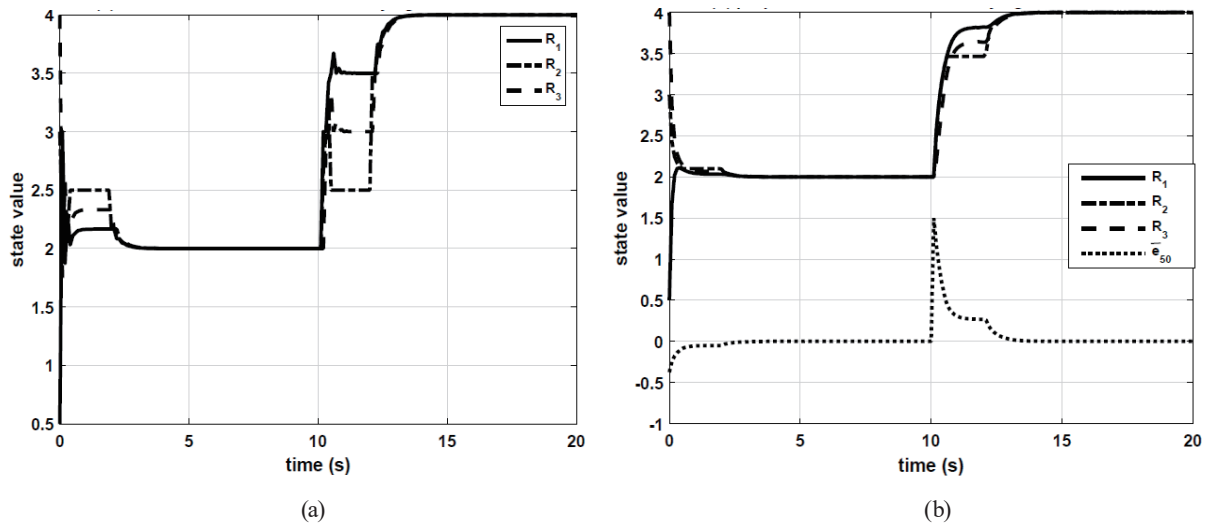


Fig. 3. State adaptation behavior according to user input change during runtime: (a) consensus and (b) proposed methods.

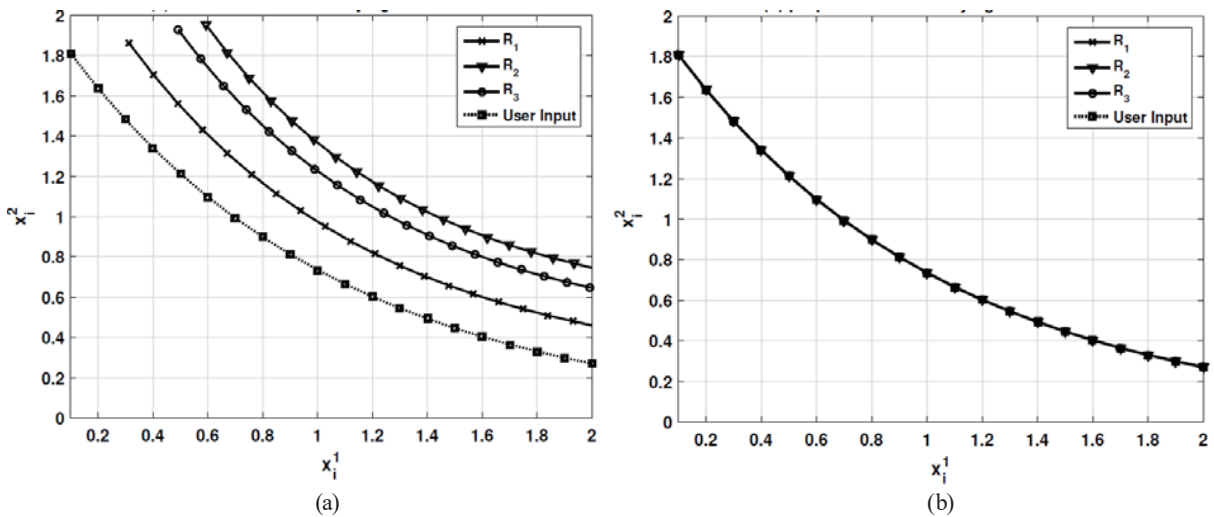


Fig. 4. State adaptation under the time-sequential control input: (a) consensus and (b) proposed methods.

large. For the proposed method, the state change between the nodes is not large despite the malfunction of R_2 and is adopted successfully according to the desired configuration. In Fig. 3(b), the dashed trajectory belongs to R_{50} controlled by the operator. The operator can estimate the state of the WSN through the value of R_{50} , which helps appropriately determine the timing for the next configuration input. The fact that e_{50} is close to 0 reflects the convergence of the WSN configuration to the control input. Therefore, the operator obtains the appropriate timing to tune the next configuration command to the WSN system by observing the e_{50} of the sink node.

Figure 4 illustrates the steady states of R_1 , R_2 , and R_3 under variable control input $\mathbf{u} = [u_1(t), u_2(t)]^T$. For simplicity, $u_1(t)$ is varied within the range of [0.1, 2.0] with 0.1 increments and $u_2(t)$

is determined as $u_2(t) = 2e^{-u_1(t)}$. The input commands are sequentially sent from the user to the WSNs. In the case of the consensus method, there is no appropriate feedback for timing to issue the next command; thus, the user command is sent periodically. The period is set to 10 s. From Fig. 4(a), the consensus method shows that the state values of the agents are changed differently and do not follow the user input exactly. The proposed method shown in Fig. 4(b) successfully controls the states of the agents as the change of the user command and the states of all agents are matched correctly by varying the user command. Also, the proposed method can estimate the time to issue the next command, which makes the control of the WSN configuration more effective when a sequential command is required by the operator.

4. Conclusions

This paper presents a scheme of parameter configuration and monitoring for WSN systems. The operator interacts with sensor nodes by applying the configuration input to a single sink node while the other nodes interact with each other according to the proposed dynamics. The proposed scheme appropriately affects the WSN operation by disseminating the operator input to the rest of the sensor nodes based on the control law of each node. The operator can effectively observe the state of the WSN by understanding dynamics and estimate the timing for the next control inputs, which is based on the feedback of the sink node only. The main contributions of the proposed algorithm over the existing schemes include the following: First, the distributed behaviors of each node, defined by the proposed scheme, automatically drive the system parameters to the desired configuration. Second, the operator can effectively observe the state of the WSN through the feedback from a single node. Finally, insights into the theoretic analysis of the proposed scheme are provided by deriving the steady states and stability of the systems.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2017R1A2B1007779).

References

- 1 J. Kulik, W. Rabiner, and H. Balakrishnan: Proc. 5th ACM/IEEE Int. Conf. Mobile Computing and Networking (1999) 174–184.
- 2 S. S. Kulkarni and M. Arumugam: Technical Report MSU-CSE-04-46, Dept. of Computer Science, Michigan State University (2004).
- 3 P. Levis and D. Culler: Proc. 11th ACM SIGOPS European Workshop (2004).
- 4 P. Levis, N. Patel, D. Culler, and S. Shenker: Proc. 1st Conf. Symp. Networked Systems Design and Implementation (2004) 2.
- 5 J. W. Hui and D. Culler: Int. Conf. Embedded Networked Sensor Systems (2004).
- 6 Z. J. Haas, J. Y. Halpern, and L. Li: IEEE/ACM Trans. Networking **14** (2006) 479.
- 7 M. Musolesi and C. Mascolo: Proc. 3rd Ann. Int. Conf. Mobile and Ubiquitous Systems (2006) 1–9.
- 8 G. Williamson, D. Cellai, S. Dobson, and P. Nixon: UKSim European Symp. Computer Modeling and Simulation (2009) 499–504.

- 9 P. Walker, S. A. Amraii, M. Lewis, N. Chakraborty, and K. Sycara: Proc. IEEE Int. Conf. Syst. Man, Cybern (2013) 2712–2717.
- 10 P. Walker, S. A. Amraii, N. Chakraborty, and K. Sycara: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2014) 1108–1113.
- 11 F. Alshahrany, M. Abbod, J. Alshahrani, and A. Alshahrani: Int. Eng. Technol. Innovation (2016) 1–15.

About the Authors



Heejung Byun received her B.S. degree from Soongsil University, Korea, in 1999, her M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2001, and her Ph.D. degree from KAIST in 2005. She was a senior researcher at the Samsung Advanced Institute of Technology and Samsung Electronics from 2007 to 2010. She is currently an associate professor in the Department of Information and Telecommunications Engineering, The University of Suwon, Korea. Her research interests include network protocol design, network modeling, controller design, and theoretical analysis.



Soo-Mi Yang received her B.S., M.S., and Ph.D. degrees in computer engineering from Seoul National University of Seoul, Korea, in 1985, 1987, and 1997, respectively. From 1988 to 2000, she was a researcher at the Korea Telecom Research Center where she worked on telecommunication networks, internet, and information security. From 2000 to 2001, she was a visiting scholar at UCLA, USA. From 2002 to 2004, she was a faculty of the Suwon Science College, Korea. She is currently an associate professor in the Department of Information Engineering at The University of Suwon, Korea. Her research interests include access control, network security, and secure system software.