Journal of Theoretical and Applied Information Technology
31st January 2018. Vol.96. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A DATA PROPAGATION METHOD CONVEYING OPERATOR INTENT FOR WIRELESS SENSOR NETWORKS

[1]HEEJUNG BYUN, [2]SUMI YANG

[1]Associate Professor, Suwon University, Department of Information and Communications, Korea

[2]Associate Professor, Suwon University, Department of Information and Security, Korea

E-mail: [1]heejungbyun@suwon.ac.kr, [2]soomiyang@hanmail.net

ABSTRACT

In this paper, we propose a data propagation method inspired by epidemic theory with aims of delay reduction and energy saving in Wireless Sensor Networks (WSNs). Also, we propose a control method for conveying operator intent to the WSNs. The proposed method controls the state of a sensor node depending on the delay requirements of a specific application and allows users to interact with the WSNs to convey desired parameter values for the operations. We analyze the proposed method based on control theory and show the convergence and stability of the WSN system. Simulation results indicate that the proposed method achieves reliable data propagation and user intent delivery while saving energy consumption in the WSN system.

Keywords: *Wireless sensor networks, Data propagation, Parameter setting*

1. INTRODUCTION

Wireless sensor networks (WSNs) connect devices that can sense and monitor physical phenomena for a wide variety of applications, such as environmental monitoring, target tracking, pervasive security, health monitoring, disaster management and recovery. Recently, delay-sensitive applications, such as, emergency and rescue applications, require application-specific functionalities and performance guarantees. Also, large scale WSNs demand a high level of decentralized optimization that achieves global performance based on each node's knowledge of its local state and local interactions with its neighbors.

The WSN systems usually need to survive for a long time. Therefore, low duty cycle has been a critical challenge to reduce the energy consumed by unnecessary idle listening [1]-[2]. However, delay-sensitive applications such as emergency and rescue applications, require their own specific functionalities for reliable data delivery. For these applications, it is important to design efficient algorithms with the aim of performance guarantee for the WSNs. Also, in order to change parameters of a WSN control algorithm, user needs to interact with the WSN. Biologically inspired modeling techniques have received considerable attention as a new way for the realization of WSNs that are robust, scalable and adaptable, yet retain their individual simplicity in computer networks [3]-[7]. Recently, many researchers have been engaged in developing biologically-inspired design paradigms WSNs while guaranteeing system robustness with individual simplicity [8]-[10], such as immune system [11], insect colonies [12], activator-inhibitor systems [13], cellular signaling systems [14], and pheromone termite model and ant colony systems [15]. In specific, epidemic-based methods have been proposed for use in data dissemination, routing, and broadcasting protocols [16]. These biologically inspired methods address the scalability for the WSNs through distributed operation, but do not guarantee that a particular global goal is achieved. In this paper, we propose a method of node state control depending on the delay requirements of a specific application and present an interaction method to convey desired parameter values for the operations.

2. RELATED WORKS

Epidemic-based communications provide an effective means of transferring data in WSNs, in which the infection transmission process corresponds to the message passing among sensors. The analogy between information dissemination in WSNs and transmission of epidemics in communities is evident: a sensor node can be referred to as infected when it receives a piece of information and stores it, and can otherwise be considered susceptible. An epidemic is regarded as a chain reaction in transmission of an infection.

Epidemic theory is the study of the dynamics on how contagious disease spread in a population, resulting in epidemic. The theory models the propagation process of an infection. Two of the simplest models describing such propagation process are the SIR (Susceptible Infected Recovered) and the SIS (Susceptible Infected Susceptible) model. In the SIR, a susceptible individual acquires infection and after a duration of time called infection period, the individual recovers and becomes immune to further infections. In the SIS, the individual becomes susceptible again after being recovered. In the general SIR model, the dynamics of the infectives (*I(t)*), susceptibles (*S(t)*), and recovered individuals (*Z(t)*) in the total population *N(t)* can be described by differential equations as follows :

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta S(t)I(t)\\ \frac{dI(t)}{dt} &= \beta S(t)I(t)-\gamma I(t)\\ \frac{Z(t)}{dt} &= \gamma I(t)\\ N(t) &= S(t)+I(t)+Z(t)\end{aligned}$$

where β and γ denote the infection rate and the removal rate of infected individuals, respectively. The SIR model states that the decaying rate of susceptible and the growth rate of infectives are calculated by the infectivity β, the number of susceptibles *S(t),* and the number of infectives *I(t)*. The total number of nodes is conserved over the interval of time taken into consideration. For the SIS model, it does not have the recovered *Z(t)* and those who are infected fall back into the susceptible.

In epidemiology, network topologies are used to study the spread of disease infections through a population, with nodes representing people and edges representing social contact. The network system is identical to the epidemic SIS model, in which an individual can be in only two state (susceptible or infected), and the change of state is a result of interaction between the individuals. The existing epidemic inspired networking protocols have focused on a modification of the SIS model, which is more realistically close to the behavior of WSNs.

One of the prominent works of data dissemination modeled by epidemic theoretical concepts is SPIN [17]. These researchers 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 TDMA based data dissemination protocol for WSN to offer a degree of reliability [18]. The Firecracker protocol [19] used a combination of routing and broadcast principles to rapidly disseminate data throughout WSNs. Trickle [20] was proposed for propagating and maintaining code updates as fast as possible to all nodes in the network. The key contribution of this protocol was `its polite gossip' that used 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 [21] added a feature that supports 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 [22]. The problem with gossip is that, if a source has very few neighbors, then the nodes will not gossip and the information dies out. Middleware has been proposed [23] to provide controlled epidemic style dissemination using information dissemination techniques to tune the process according to the desired reliability. In another case [24] an MRO (multi-rumor overwriting) model was proposed. The model described dissemination and interaction of periodically sensed data, which were distributed by gossiping, without increasing the traffic on the network.

However, these existing algorithms have not achieved satisfying a desired level of performance of the delay-sensitive application. In order to convey user intent to the WSN, a centralized method was explored [25]-[26]. More decentralized approaches include methods to allow the user to adjust the autonomy of a small subset of nodes to influence the WSN behavior [27]. However, as the number of nodes involved in the task grows large, such approaches can become less effective and unstable due to a lack of scalability.

To address these concerns, we design an epidemic-inspired data propagation method with adaptive information spread for dynamic WSN environment. Also, we provide a user control of WSNs by conveying user intent or parameter changes for their operation.

3. EPIDEMIC INSPIRED DATA PROPAGATION METHOD

2.1 Model

We consider a WSN consisting of N nodes. Let $\mathbf{N}=\{1,2,...,N\}$ denote the set of sensor nodes in a WSN. Each sensor node can be only in one of two states; active or sleep. Sensor nodes turn off their radios when asleep and turn them on when active to exchange information packets. We formulate a discrete dynamic model as follows:

$$
\begin{aligned}
N_s(n+1) &= \max(\min(N_s(n)-\alpha(n)N_a(n)N_s(n) \\
&\quad +\beta N_s(n), N_t(n)-N_{a,\min}), 0) \\
N_a(n+1) &= \min(\max(N_a(n)+\alpha(n)N_a(n)N_s(n) \\
&\quad -\beta N_a(n), N_{a,\min}), N_t(n)) \\
N_t(n) &= N_s(n)+N_a(n),
\end{aligned} \tag{1}
$$

where $N_s(n)$ and $N_a(n)$ are the number of nodes in sleep state and in active state at time slot n, respectively, α is the infectivity rate, β is a control parameter, and N_t is the total number of sensor nodes. The value of $N_{a,min}$ is the minimum of N_a needed to ensure a certain level of reliability, and can be determined based on the desired delivery ratio.

Eq. (1) states that the number of active nodes are calculated with dynamic infectivity α. Existing epidemic algorithms have used predefined infectivity and have not been used to control infectivity according to environmental change. To solve this problem, our proposed algorithm controls the value of α according to the desired performance of an application, and to network condition as well.

We denote $d(n)$ and d_r as the maximum measured delay among nodes and the delay requirement of the application, respectively. The sink node measures the maximum delay at every control period and notifies the sensor nodes of the measured delay by piggybacking on an ACK packet. With this feedback information, the dynamics of α is modeled by the following equation:

$$
\begin{aligned}
\alpha(n+1) &= (1-\varepsilon)\alpha(n)+\eta\Phi(\sigma(n)) \\
\sigma(n) &= d(n)-d_r,
\end{aligned} \tag{2}
$$

where $\varepsilon(<<1)$ is a positive constant and η is the control parameter to be chosen. The function Φ is an increasing function from the interval $[-d_M, \infty]$, were $-d_M$ is the upper bound of delay requirement.

The combined model of (1)-(2) determines that when the measured delay exceeds the delay requirement, the value of α increases, which leads to new active nodes among the total nodes. Alternatively, when the measured delay is less than the delay requirement, the value of α decreases, resulting in an increased number of of sleeping nodes. Using the value of N_a determined by (1) and (2), each sensor node decides whether it will have active or inactive status for the next controller time slot.

To specify the detailed node scheduling algorithm, we introduce a random value ω, following the uniform distribution within [0, 1]. Each node independently generates a random value. If the ratio of the active node to the total sensor nodes (N_a/N_t) is less than ω then the node goes to sleep. On the other hand, if the active node ratio is greater than ω, the node will be active during the next controller time slot. If the maximum measured delay exceeds the required delay, the values of α and N_a increase, resulting in a higher probability of being active.

This leads to delay reduction and more frequent packet transmission, but if the maximum measured delay is smaller than the required delay, the value of α and N_a decreases, resulting in energy savings by increasing sleep time. While active, if the sensor node has not received any packet during k-straight time slots, the value of measured delay d is written as 0, resulting in decrease of N_a. If N_a reaches N_{min} while d is still 0, then the sensor node goes back to the initial state and wakes up every I_M. Consequently, we can see that for a given level of application requirement and network condition, the proposed scheme provides an efficient way to control the infectivity α and the number of active nodes N_a, which leads to adaptive node scheduling according to the application-specific requirements.

2.2 Conveying parameter changes to the WSNs

The proposed model depends on some parameters such as β, d_r, for its operation. Theses parameters should be properly conveyed from the operator to the WSNs. In this paper, we focus on indirect parameter setting aided by an autonomous algorithm.

To establish control model of a user-WSN interaction, we consider a system with inputs controlled by a user, that is, the user controls the behavior of the WSN by interacting with sensor nodes. Let $u(t)$ be the input to the system at time t specific, $x(t)$ represents the states of all sensor nodes at time t and can be indicated as vector $\mathbf{x}(t)$ =[$x_1(t)$,

$x_2(t)$, ..., $x_N(t)]^T$, where $x_i(t)$ is the state vector of sensor node i at time t. We denote N_i as the set of neighbor nodes of node i. The user interacts with the WSNs by applying control input to a single node (gateway node), while the other nodes interact with each other according to the proposed dynamics. With this respect, we propose the dynamics of the system as follows:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3}$$

where

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

We consider that input function $u(t)$ is piecewise constant having a finite linear combination of indicator functions of intervals. The user input is directly applied for N-th node by letting $x_N(t) = u(t)$. Then the dynamic for the N-th node is as follows:

$$\dot{x}_N(t) = 0 \tag{4}$$

According to (4), user directly controls the state of a single gateway node and the interactions between the nodes are handled autonomously. Consequently, the user takes actions independent of the number of nodes and supervises the WSN system as a single entity, resulting in a control complexity of $O(1)$.

Let us denote $x_s = [x_{1s}, x_{2s}, ..., x_{Ns}]^T$ as a steady state vector $x(t)$, where x_{is} is the steady state of node i. For a desired user input $u(t) \approx u$, the steady states of x_i is derived as follows:

$$\begin{aligned} x_{is} &= \frac{1}{N_i}(1_{(i,1)}x_{1s} + 1_{(i,2)}x_{2s} + \cdots + 1_{(i,N)}x_{Ns}) \\ x_{Ns} &= u \end{aligned} \tag{5}$$

That is, the steady states are rewritten as

$$Ax_s = b \tag{6}$$

Where

$A=$

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

$b=$

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

The indicator function $1_{(i,j)}$ is defined as

$$1_{(i,j)} = \begin{cases} 1 & \text{if } j \in N_i \\ 0 & \text{else} \end{cases}$$

Then, the steady state xs is derived as follows:

$$x_s = A^{-1}b = [u \;\; u \cdots u]^T \tag{7}$$

When the system is asymptotically stable, the trajectory will converge to the steady state as time goes to infinity. In order to ensure the stability of the proposed system, we denote
$y(t) = [y_1(t), y_2(t), ..., y_N(t)]^T$ where $y_i(t) = x_i(t) - x_{is}$. Then, the state space model for control design is written as follows:

$$\dot{y}(t) = A_s y(t) \tag{8}$$

where

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

With the equilibrium point y_s, we define $e = y - y_s$ as the error. Then the error dynamics can be described as

$$\dot{e} = A_s(e + y_s) \tag{9}$$

We use Lyapunov Theory to claim that the system defined by (9) will be convergent if we can find a Lyapunov function $V(e)$ that satisfies the following conditions:

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

We assume that there exist positive definite matrices P and Q, $P = PT$, such that

Journal of Theoretical and Applied Information Technology
31st January 2018. Vol.96. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$PA_s + A_s^T P = -Q. \quad (10)$$

TABLE 1: DEFAULT PARAMETERS USED IN SIMULATIONS

Parameter Name	Default Value
simulation area	100m × 100m
number of nodes	100, 200
transmission range	20m
carrier sense range	40m
duty cycle (wakeup interval)	1s
packet generation interval	60s
current consumption: Tx	17.4mA
current consumption: Rx	19.7mA
current consumption: mode switch	10.05mA
mode switch time	300μs
backoff time	30ms

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

$$\begin{aligned}\dot{V}(e) &= \frac{1}{2}\left(\dot{e}^T P e + e^T P \dot{e}\right) \\ &= \frac{1}{2} e^T (A_s^T P + P A_s) e \\ &= -\frac{1}{2} e^T Q e \leq 0\end{aligned}$$

Based on the above analysis, both conditions of the Lyapunov function $V(e)$ have been satisfied, thus we can claim that the system is stable and the error dynamics will converge to zero exponentially by satisfying the condition of (10) on the system parameters.

This result guarantees the autonomous adaptation of node state according to user-defined swarm configuration, resulting in asynchronous convergence of the swarm to the desired global behavior.

4. SIMULATION RESULTS

The simulations are conducted using a simulator written in MATLAB. The simulator captures real life events such as carrier sensing, backoff, and collisions. At the beginning of a simulation, nodes are randomly placed inside the simulation area, and the source node is placed at the center. Once nodes are deployed, nodes start duty-cycling by switching between active and sleep mode. At some point, the source node starts generating

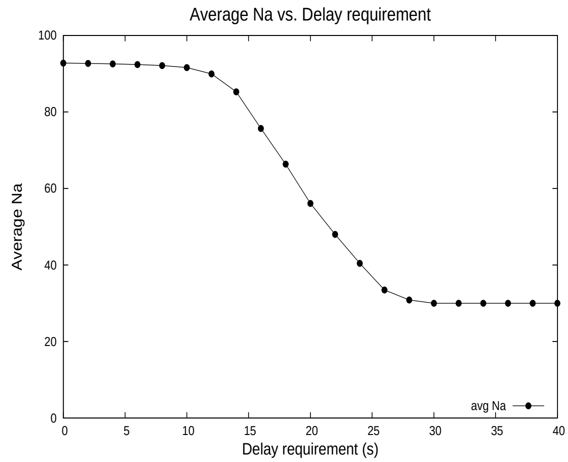

Figure 1: Average Na varying delay requirements.

packets periodically at a fixed rate. The packets are stamped with sequence numbers, so that a node does not receive or forward duplicate packets.

The simulation area is 100m × 100m, where the entire network is divided into equally shaped grids, and the nodes are uniformly deployed. We set N = 100, and the nodes are arbitrary connected. The gateway node is s chosen randomly by the operator. The channel capacity is set to 200 kbps, the transmission range and carrier sense range to 20m and 40m, respectively. The goal of the source node is to send the packet to all the other nodes in the network, within a given delay. The default parameters used in the simulations are listed in Table 1. These parameters are used unless other specified. The parameter N_a decides the average number of nodes that wake up at a particular active period. If N_a is too low, the packet will die out before being disseminated throughout the entire network.

Fig. 1 shows the average of N_a varying delay requirements. We can observe that the number of active sensor nodes is successfully controlled by the delay requirement. In specific, as the delay requirement becomes larger, the value of N_a decreases, which makes the number of sleep nodes increase. On contrary, the value of N_a increases when the delay requirement becomes stricter.

Fig. 2 shows the trajectories of state vectors of node 1, node 2, and node 3. We change parameter setting [β, d_r] by letting $x_N(t) = u(t) = [x=4, y=4]^T$. In spite of the different initial values of parameter state vector, the value of parameter state of each node converges to the desired global configuration, which

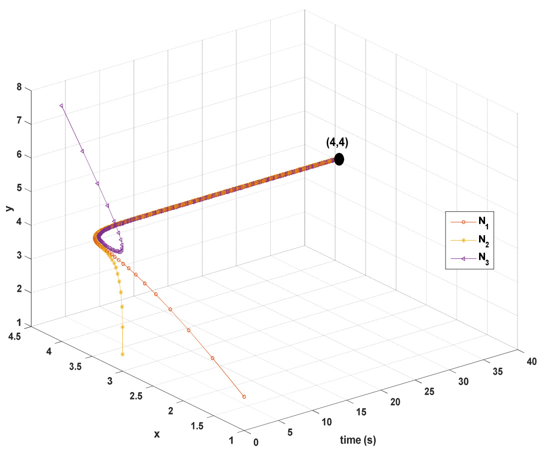

Figure 2: Parameter changes with node interaction, u = [4,4]^T.

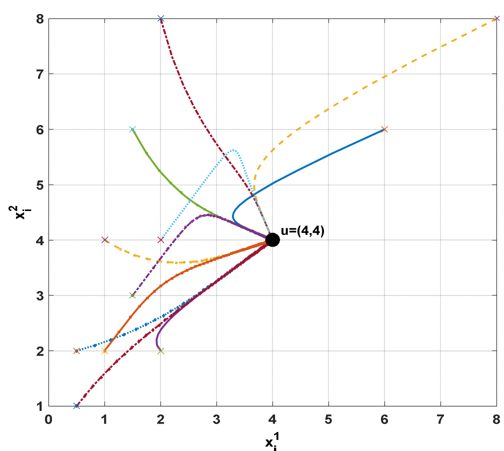

Figure 3: States of 10 nodes. The initial state of each node is indicated as ×.

is set by the user. Also, by guiding the leader node to the desired configuration, the user can change the parameter of the algorithm of all sensor nodes autonomously.

Fig. 3 shows the trajectories of states of 10 among 100 nodes ($R_1, R_2, \cdots, R_{10}$). The initial state of each node is set differently and indicated as ×. Results indicate that the state of each node is autonomously adapted and converged to the control input given by the operator.

Next, we examine the trajectories of states corresponding to control input changes during the runtime. The user input is initially set to $u = 2$ and changed to $u = 4$ at 40s. Fig. 4 shows the state

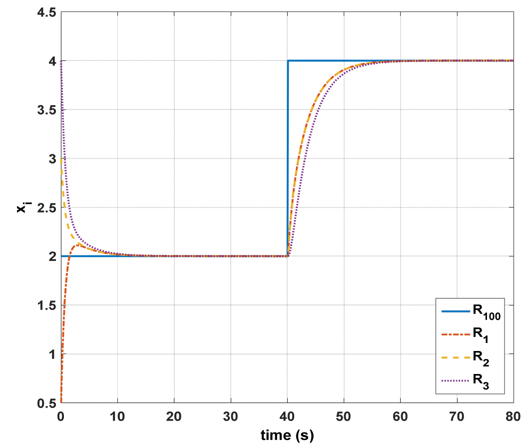

Figure 4: State adaptation according to user input change during runtime, u=2 and u=4.

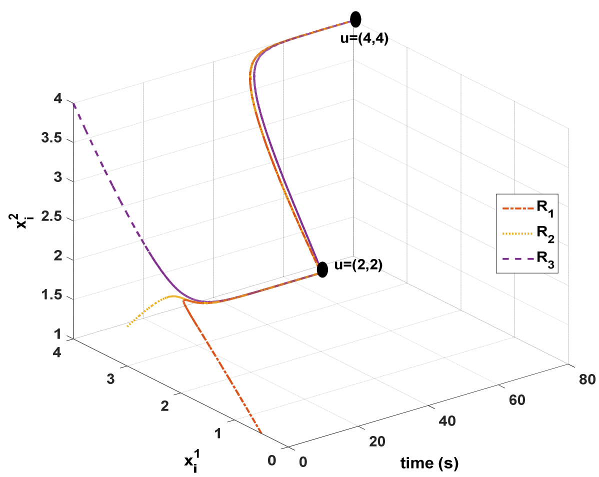

Figure 5: State adaptation according to user input change during runtime, u=[2,2]^T and [4,4]^T.

adaptation behavior according to control input changes. The solid trajectory belongs to the leader node (R_{100}) controlled by the human operator. Fig. 4 shows each node starting with a different initial value, but after 10s, all nodes converge to the same value each other. In 40s, the control input changes to 4, and after 10s, the state value of the node also converges to 4. Therefore, the human operator can tune the configuration parameters of nodes by guiding the leader node to the desired state. Furthermore, the operator can know the state of the entire system through the value of R_{100}, which helps

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in properly determining the timing for the next control input.

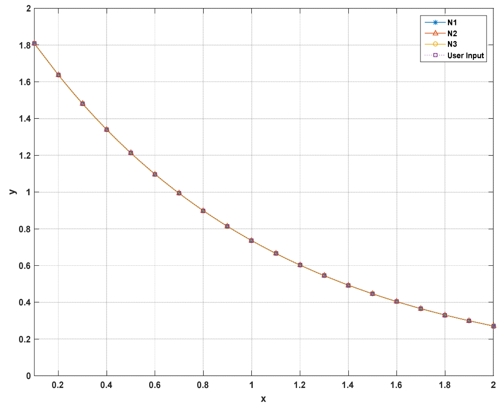

Figure 6: Parameter changes varying user inputs.

Next, the control input $u = [2,2]^T$ is first applied and then changed to $u = [4,4]^T$ at 40s. Fig. 5 shows the user input change. Similar to previous results, the state value of each node successfully adjusts according to the operator's input change and shows convergence of the entire swarm to the desired state.

Fig. 6 shows the parameter changes varying user inputs. The graph represents averaged state vector. We can observe that the values of state vector of each node are successfully converged to the user inputs. Therefore, the combinations of data propagation algorithm and user intent conveying algorithm would be effective in modifying the behavior to achieve the desired configuration with such a WSN system.

We show the impact of the control parameter η. In Figs. 7-8, the time behavior of N_a is illustrated in case of $\eta = 0.001$ and $\eta = 0.1$. From Fig. 8, we observe that when the values of η is set as 0.1, the behavior of N_a oscillates excessively, resulting in unstable system performance. The selection of system parameters can be considered for the stable operation of the proposed system. These issues will be the subjects of our future work.

5. CONCLUSIONS

We propose a data propagation method based on the epidemic theory for delay reduction and energy saving in WSNs. Also, a method of conveying parameters to the sensor systems from the operator is proposed. From the simulation results, we observe that the proposed algorithm achieves data

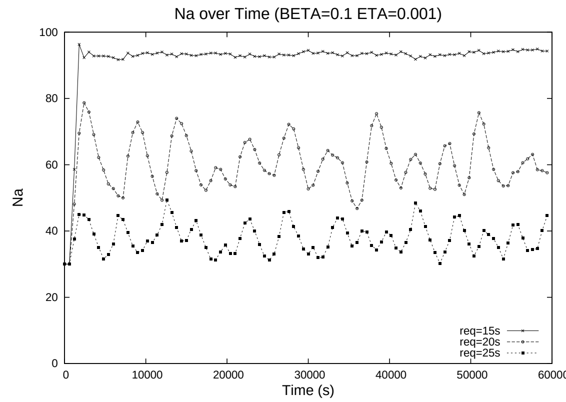

Figure 7: Impact of the control parameters β and η: β =0.1, η = 0.001.

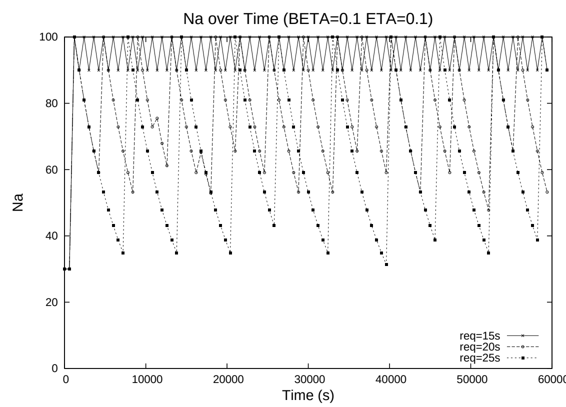

Figure 8: Impact of the control parameters β and η: β =0.1, η = 0.1.

propagation with user intent while saving energy consumption.

An important area for further study includes leader node election, estimation of performance impact of applied parameters, and decision making for the next appropriate command. This research could also investigate the effective collaboration among nodes for appropriate decision making.

ACKNOWLEDGEMENT

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

REFRENCES:

[1] W. Ye, J. Heidemann, and D. Estrin, "An energy efficient mac protocol for wireless sensor networks," *IEEE INFOCOM*, pp.1567-1576, 2002.

[2] Bachier, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222-248, 2010.

[3] F. Dressler and O.B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881-900, 2010.

[4] T. Nakano, "Biologically inspired network systems: A review and future prospects," *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 41, no. 5, pp. 630-643, 2011.

[5] C. Charalambous and S. Cui, "A biologically inspired networking model for wireless sensor networks," *IEEE Network*, vol. 24, no. 3, pp. 6-13, 2010.

[6] K. Leibnitz and M. Murata, "Attractor selection and perturbation for robust networks in fluctuating environments," *IEEE Network*, vol. 24, no. 3, pp. 14-18, 2010.

[7] C. Cheng, C. K. Tse, and F. C. M. Lau, "A bio-inspired scheduling scheme for wireless sensor networks," *IEEE Vehicular Technology Conference*, pp. 223-227, 2008.

[8] C. Charalambous and S. Cui, "A biologically inspired networking model for wireless sensor networks," *IEEE Network*, vol. 24, pp. 6-13, 2010..

[9] F. Dressler and O.B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881-900, 2010.

[10] T. Nakano, "Biologically inspired network systems: A review and future prospects," *IEEE Trans. On Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 41, no. 5, pp. 630-643, 2011.

[11] B. Atakan and O. B. Akan, "Immune system based distributed node and rate selection in wireless sensor networks," *Int. Conf. IEEE/ACM Bio-inspired Models Network, Information, Computing Systems,* pp. 1-8, 2006.

[12] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, 2007.

[13] F. Dressler, "Self-Organized Event Detection in Sensor Networks using Bio-inspired Promoters and Inhibitors,*" Int. Conf. ACM/ICST Bio-Inspired Models of Network, Information and Computing Systems*, 2008.

[14] T. Iwai, N. Wakamiya, and M. Murata, "Error-tolerant coverage control based on bio-inspired attractor selection model for wireless sensor networks," *Int. Conf. IEEE Computer and Information Technology*, pp. 723–729, 2010.

[15] G. Singh and S. Bal, "A Review of Ant Colony System Algorithm and its Models," *International Journal of Advanced Research in Computer Science*, vol. 24, no. 4, pp. 266-269, 2017.

[16] Pradip De and Sajal K. Das, Y. Gao, and D. Zuo, "Epidemic models, algorithm and protocols in wireless sensor and ad-hoc networks," *John Wiley & Sons*, doi: 10.1002/9780470396360.ch3, 2008.

[17] J. Kulik, W. Rabiner, H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," *Proc. the Fifth ACM/IEEE Int. Conf. Mobile Computing and Networking,* pp. 174-184. 1999.

[18] S. S. Kulkarni and M. Arumugam, "INFUSE: a TDMA based data dissemination protocol for sensor networks," *Technical Report MSU-CSE-04-46*, Dept. of Computer Science, Michigan State University, 2004.

[19] P. Levis and D. Culler, "The firecracker protocol," *Proc. the 11th ACM SIGOPS European workshop*, 2004.

[20] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," *Proc. the 1st conf. Symp. Networked Systems Design and Implementation*, vol. 1, pp. 2-2, 2004.

[21] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," *Int. Conf. Embedded Networked Sensor Systems*, 2004.

[22] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Networking,* vol. 14, no. 3, pp. 479-491, 2006.

[23] M. Musolesi and C. Mascolo, "Controlled epidemic-style dissemination middleware for mobile ad hoc networks," *Proc. 3rd Ann. Int. Conf. Mobile and Ubiquitous Systems*, pp. 1-9, 2006.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[24] G. Williamson, D. Cellai, S. Dobson, and P. Nixon, “Modelling periodic data dissemination in wireless sensor networks,” *UKSim European Symposium on Computer Modeling and Simulation*, pp.499-504, 2009.

[25] Z. Kira and M. A. Potter, “Exerting human control over decentralized robot swarms,” *Proc. Of the Intl, Conf. on Systems, Man, and Cybernetics*, 2010.

[26] X. C. Ding, M. Powers, M. Egerstedt, S. Young, and T. Balch, “Executive decision support: single agent control of multiple UAVs,” *IEEE Robotics and Automation Magazine*, 2009.

[27] S. Bashyal and G. K Benayagamoorthy, “Human swarm interaction for radiation source search and localization,” *IEEE Swarm Intelligence Symposium*, 2008.