INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(1):90-102, February 2017.

Initial Phase Proximity for Reachback Firefly Synchronicity in WSNs: Node Clustering

M. Misbahuddin, R.F. Sari

Misbahuddin Misbahuddin*
Department of Electrical Engineering, Faculty of Engineering,
Universitas Indonesia, Depok, 16424, Indonesia
*Corresponding author: misbahuddin@ui.ac.id

Riri Fitri Sari
Department of Electrical Engineering, Faculty of Engineering,
Universitas Indonesia, Depok, 16424, Indonesia
riri@ui.ac.id

Abstract: Synchronicity is one of the essential basic services to support the main duties of Wireless Sensor Networks (WSNs). Synchronicity is the ability to arrange simultaneously collective actions in WSNs. A high-rate data sampling to analyze the seismic structure and volcanic monitoring is the important applications requiring a synchronicity. However, most of the existing synchronicity algorithm is still executed in a flat network, so that it requires a long time to achieve a synchronous condition. To increase the convergence rate, the synchronicity can be executed concurrently through a clustering scheme approach. In this work, the such scheme is called as the Node Clustering based on Initial Phase Proximity for Reachback Firefly Synchronicity (NCIPP-RFS). The NCIPP-RFS solves in three steps: (1) constructing the node clustering, (2) intra-cluster synchronicity, and (3) inter-cluster synchronicity. The NCIPP-RFS method is based on the firefly-inspired algorithm. The fireflies are a species in the natural system, which are able to manage their flashing for synchronicity in a distributed manner. The NCIPP-RFS was implemented in NS-3 and evaluated and compared with Reachback Firefly Algorithm (RFA). The simulation results show a significant increase in the convergence rate. The NCIPP-RFS can reach a convergence time shorter than the RFA. In addition, the NCIPP-RFS was compared in the various numbers of clusters, where the least number of clusters can reach the fastest convergence rate. Finally, it can also contribute significantly to the increase of the convergence rate if the number of nodes is greater than or equal to 50 nodes.
Keywords: wireless sensor network, synchronicity, node clustering, phase proximity, firefly-inspired algorithm.

1 Introduction

Wireless Sensor Networks (WSNs) is a set of spatially distributed autonomous sensor nodes that could to interact locally, and some nodes of them interact with a sink node or Base Station. A sensor node in WSNs is a device that has the limitations in the processing unit, communication resources, and sensing capabilities. The WSNs has been implemented in various applications for environmental monitoring, condition sensing, and process automation such as battlefield surveillance, habitat monitoring, coordinated target detection and localization, chemical attacked detection, ubiquitous healthcare, and home automation. The duties of the WSNs can be executed properly when it is supported by robust basic requirements such as time synchronization, synchronicity, self-configuration, and self-localization. For example, time synchronization is extremely required to ensure the high accuracy measurement for an event-driven measurement application in an area where an event is detected [1]. Synchronicity inspired by the biological principle of a firefly is adopted by [2] for handling a dynamic node clustering in data readings. Self-configuration characteristic for a routing protocol of WSNs developed by [3] aimed at discovery the best route for delivering information with minimum energy consumption. Finally, self-localization is a basic feature in WSN for finding out the location of unknown nodes because GPS that is usually used to find the location of a device is not suitable for WSNs [4].

Many phenomena exist in the natural system around us, which have inspired many scientists to solve various problems within the engineering field or the specific problems of WSNs. The basic requirements of WSNs such as synchronization and synchronicity need a self-organized way. Both are a mutually complementing requirement of WSNs. The synchronization is an ability to align the time of node's internal clock referring on the global time to perform a simultaneously collective action. On the other hand, the synchronicity is a way to align the phase of the internal clock of node to conduct a synchronously collaborative action. The application of the synchronicity in WSNs is very useful as a simple sensor network coordinator in sampling the high data rates such as seismic analysis of structure [5], and volcanic monitoring [6]. Moreover, is also used as a scheduling mechanism of the node duty cycles in order to save energy, so that all nodes in a network can wake up at the same time.

There are some challenges of the robust synchronicity requirements, i.e. simple, fast, low energy consumption, self-configuration and high scalability. In this study, we propose a new node-clustering synchronicity method using a firefly-inspired algorithm to address three synchronicity requirements in simple, fast,and self-configured way.

Clustering in the WSNs is often used in some applications because it is extremely useful for various purposes. The purposes are divided as primary and secondary [7]. The first purpose represents the objectives that are the most substantial in the node clustering such as scaling, fault-tolerance, data aggregation, load balancing, network topology stabilization, network lifetime extension. Instead, the second purpose points out the objectives that are not highly important, and they are indirectly achieved by clustering node such as increasing connectivity, reducing routing delay, avoiding collision, and utilizing sleeping schemes. Therefore, in this research we propose the node clustering model as a new approach to synchronize the node in the network, where the node clustering is an extremely important requirement to relieve the load of the network to reach the synchronicity.

Our NCIPP-RFS approach provides two main contributions that consist of (i) in clustering, there are some node subsets that perform an intra-cluster synchronicity in parallel, continued to an inter-cluster synchronicity. The total of periods required to execute both synchronicity processes is smaller than that of without clustering. This emphasizes that our approach is faster than the non-clustered synchronicity algorithm. (ii) Self-configuration is an important problem in the distributed system for executing both its primary duties and basic function as well as for the synchronicity function. In fact, there are many natural phenomena around us in which their population can organize themselves toward a synchronicity state, in which one of them is the firefly. The behavior of firefly flashing has inspired a number of researchers to create the firefly-inspired synchronicity algorithms [8]-[13]. In this research, we utilize the synchronicity algorithm developed by [10] for intra-cluster synchronicity.

The remainder of this paper is organized as follows: Section 2 presents literature review related with the firefly-inspired synchronicity. Section 3 describes the approach used to solve the synchronicity requirements. Section 4 presents the simulation results to show the performance evaluation. Finally, Section 5 concludes this paper and ideas for future work.

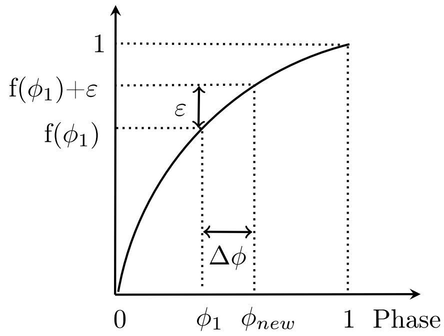

Figure 1: The state function of M&S model

2 Firefly-inspired Synchronicity

In the natural system, there are several synchronicity phenomena that have been observed by many researchers to understand the mutual interaction of a population toward a self-organized synchronicity without a notion of time. Examples of the natural synchronicity include circadian rhythm [14], pacemaker cell of the heart [15], and synchronous flashing of fireflies [16].

Fireflies are one of the species around us, which can interact mutually to fire synchronously. This phenomenon is one of the most spectacular self-organized synchronicity, which was imitated as a firefly-inspired synchronicity algorithm. The population of the fireflies can be analogized as a population of the pulse-coupled biological oscillators (PCO) that was introduced by Mirollow and Strogatz [8], which is known as the M&S model. However, this model cannot be implemented directly in a real WSNs because it still uses some ideal assumptions that are not in accordance with the realistic wireless communication [17]. The assumptions are: (1) the characteristic of oscillators is identical, (2) the node's firing event occurs and other nodes respond instantaneously. (3) Node's computations are conducted perfectly and immediately.

The M&S model presents a basic concept of the firefly-inspired synchronicity algorithm, which is described through two pulse-coupled oscillators. Each oscillator is characterized by a monotonically increasing and concave down function representing a firing function as shown in Figure 1. When the oscillator's phase increase monotonically to reach a threshold, it fires and falls immediately to zero. The mutual interaction occurs between two oscillators when an oscillator fires and sends a firing message to another oscillator that causes another oscillator responds by adjusting its own phase toward firing.

As a result, it will jump to a new phase ϕ_{new} with coupling strength ε. The new phase can be calculated using the following equation [6]:

$$\phi_{new} = \min(1, f^{-1}(\phi) + \varepsilon)) \tag{1}$$

where f(ϕ)=$\frac{1}{b}$.ln(1+[e^b-1].ϕ), and b is a dissipation parameter.

The essential weakness of the M&S model, which violates the practical WSNs, is when responding immediately a firing message that is sent by its neighbor nodes without considering an unpredictable delay because of the channel contention prior to message transmission. Therefore, the Reachback Firefly Synchronicity (RFA) developed by Werner-Allen et al.[10] overcomes the realistic wireless communication problems of the PCO model. Three problems handled by the RFA are related the wireless communication and one problem of the load computation. They are the timestamping message, the notion of pre-emptive message staggering, reachback response, and simplified firing function. The amount of time a message that was delayed before being broadcasted can be estimated using the low-level timestamping. The oscillator of the PCO model that reacts shortly to each firing event can be overcome utilizing the notion of reachback response, that is, all firing events as the phase jumps are recorded and are calculated once at the end of each period that is used to jump at the beginning of the cycle. The wireless contention, in the worst case, can be avoided employing the notion of pre-emptive message staggering. Finally, the computation complexity is reduced by simplifying the firing function. The detail description of four ways to be applied in the real wireless network are as follows:

1. A node experiences a delay between when it reaches a firing and when it starts to transmit a message. The delay can be estimated using MAC-layer timestamping. The measurement of the MAC-delay can be started using a trigger to record it in the header of the outgoing message when the message is transmitted. In the receiver node, the information is used to determine the proper firing time by calculating the difference between the MAC-delay and the reception time of the message.

2. In the M&S model, a node responds immediately to each firing message from other nodes. In contrast, the RFA uses the notion of reachback response to record each received firing message and calculates them as the phase jumps once at the end of each period, which is used as a jump at beginning of the next cycle as illustrated in Figure 2. This approach will be discussed in more detail in subsection 3.2.

3. In the M&S model, perhaps many nodes transmit the firing message together when they fire simultaneously. This event is the worst case of the CSMA scheme because they can cause channel collisions. To avoid such a worst case and to control the extent of the message delay, the RFA model introduced a notion of pre-emptive message staggering by adding a random transmission delay to the firing message at the application level of the node. The delay value is assigned to a uniformly random value between 0 up to a constant D before node fires. Furthermore, a random waiting time W (where D<W<<T) is added for a node that has fired to receive the delayed messages before processing the firing message in the queue.

4. The M&S model's firing function can be simplified to make a faster calculation of firing response. Therefore, the RFA algorithm reduces the computation complexity by simplifying the firing function. The new phase obtained after receiving a firing message at phase ϕ_1 is $\phi_2 = \phi_1 + \varepsilon\phi_1$ or $\phi_2 = (1 + \varepsilon)\phi_1$.

3 The Proposed Method of Node Clustering Based on Phase Proximity for Firefly-inspired Synchronicity

The challenges of synchronicity in the wireless sensor network are how to increase the convergence rate, and to handle the latency delay in wireless communication. Both challenges aim to reduce the power consumption. The model proposed by Sun Yi et al. [18] is a centralized node clustering mechanism through sorting the phase value of nodes. This model is difficult to be implemented in WSNs because they require a node as coordinator to gather the phase value of all nodes. Furthermore, the coordinator has to be able to organize the phase order of all nodes in order to construct the node clustering. This research proposes a new decentralized node clustering mechanism based on original phase proximity of nodes where the nodes in the

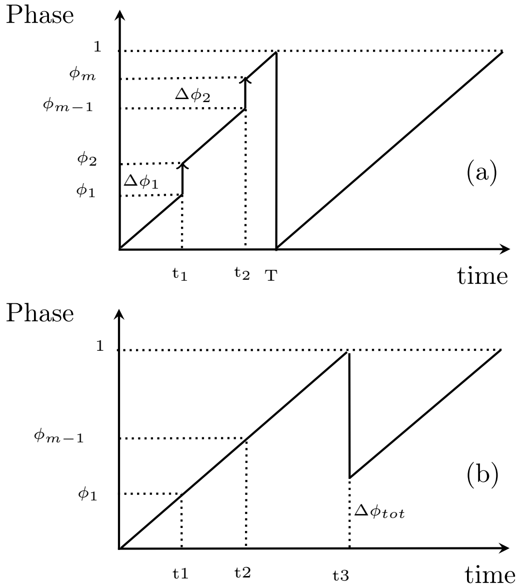

Figure 2: Phase jump in (a) M&S model, and (b) RFA model

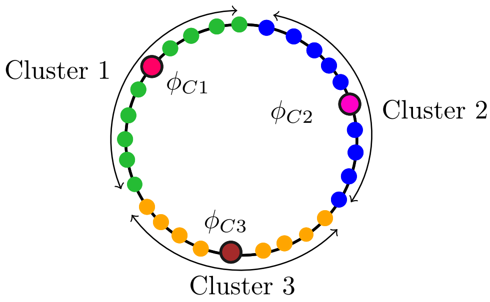

Figure 3: Illustration of the phase cluster for three clusters

network construct the clusters through a self-configuration way. This approach is called Node Clustering based on Initial Phase Proximity for Reachback Firefly Synchronicity (NCIPP-RFS).

Generally, synchronicity steps of the NCIPP-RFS are similar to the CFSA method: cluster-construction, intra-cluster synchronicity, and inter-cluster synchronicity. However, unlike CFSA method, NCIPP-RFS uses a new approach in the cluster construction. The clusters are constructed based on the original phase value proximity of nodes that are organized in a self-configuration way without assistance a node coordinator. Moreover, the inter-cluster synchronicity is added with a Late Sensitive Window (LSW) factor to avoid the overshoot of jump in a group or cluster.

3.1 Construction of the Clusters Based on Phase Proximity

Node Clustering Based on Initial Phase Proximity (NCIPP) is a simple algorithm used for clustering in WSNs. This algorithm partitions the nodes into k clusters. The node clustering is constructed using the following steps:

1. Arbitrarily assign k points as the center clusters, k being the number of clusters desired. The assignment of the center point of the cluster is intended to ensure that each node only

exists in a cluster. Next, each node will internally assign its own cluster by calculating a nearest distance to every center point of the cluster (ϕc_1, ϕc_2, ϕc_j, ...,ϕc_k). The distance can be computed by the following Euclidean equation:

$$d\phi_{ij} = \sqrt{\phi_i^2 - \phi c_j^2} \tag{2}$$

where $d\phi_{ij}$ is the distance between the i^{th} node's phase ϕ_i and the j^{th} center point of the cluster ϕc_j, and the node's phase is $\phi_i \in [0\ 1]$. Each node in the network executes the cluster-initialization function internally to determine its own cluster. The pseudo code is shown in Algorithm 1 in which the cluster number can be obtained through the minimum value of the clusterArray. The center points are determined based on the number of clusters k that is desired, where they are assigned through a balanced division of the distribution of phase values (between 0 and 1) in k groups. For illustration, let's assume that there are three clusters ($k = 3$) where the center point of the clusters are denoted by $\phi c_1 = 0.167$, $\phi c_2 = 0.5$, and $\phi c_3 = 0.833$ respectively. When the uniformly distributed phase values are mapped in a circle window in which the zero and one phase coincide in a point, the nodes will form three clusters as shown in Figure 3.

2. Each node provides an internal data structure called neighborNodes to store the source address (srcAddr) of its neighbor nodes in the same cluster and has a cluster number variable (noCluster).

3. The node clustering is executed by two main functions, i.e., the BroadcastIDCluster and the RecieveIDCluster. The pseudo codes of the both functions are shown in Algorithm 2. The *BroadcastIDCluster* is a function to broadcast the source address, and the cluster number of the transmitter node to other nodes in the same cluster. To avoid simultaneous transmission in the periodic broadcast, it is added with a random interval delay and a beacon expire timer in the *BroadcastIDCluster*. On the contrary, the *ReceiveIDCluster* is a function to receive the broadcasting messages. If the sender's cluster number is equal to the receiver's cluster number, the receiver node stores the sender's address and cluster number into its data structure, the *neighborNodes*. This means that they are in the same cluster.

Algorithm 1 Cluster initialization

ClusterInitilization()
Assign the number of cluster k
Assign the center point of the clusters (ϕc_1, ϕc_2, ...,ϕc_k)
for j:k **do**
    clusterArray[j] $\leftarrow d\phi_{ij}$
noCluster $\leftarrow$ index of min(clusterArray)

3.2 Intra-cluster Synchronicity

After all logical clusters based on the initial phase are constructed. Every logical cluster can start to perform a synchronicity concurrently. There are two firefly-inspired algorithms that can be used to perform the synchronicity, i.e., M&S and RFA model. The M&S model still uses a simple algorithm without considering the delay effect of real communication in WSNs. A firing

Algorithm 2 Node Clustering

```
BroadcastIDCluster(srcAddr, noCluster)
Send(srcAddr, noCluster)
Delay(interval+rand())
BeaconExpireTimer ()

ReceiveIDCluster(srcAddr, noCluster)
if (noClusterOfSender=noCluster) then
    neighborNodes.srcAddr ← srcAddr
    numCluster ← numCluster+1
```

message that is sent by a node is responded instantaneously by its adjacent nodes regardless of its unpredictable delay because of the delay latencies [8]. For illustration, let's assume that in a cluster, node x receives a firing message from another node y at time t_1 or phase ϕ_1. In response, the node x adjusts its oscillator's phase by increasing slightly to a new phase ϕ_2 that can be computed through the following firing function:

$$\phi_2 = \min(1, (1 + \varepsilon)\phi_1) \tag{3}$$

Every time node x receives a firing message from any node in same cluster; it calculates its phase jump using the following equation:

$$\Delta\phi_i = \min(1, (\phi_2 - \phi_1)) = \min(1, \varepsilon\phi_1) \tag{4}$$

To address the delay latency in a real network, a reachback response mechanism of the RFA model is a proper choice. In this model, the node x does not jump every time it receives a firing message, but it always stores its entire phase jumps until the period end. Furthermore, the sum of phase jumps in the previous period will be used to jump at the beginning of next period. Figure 2a illustrates the M&S model where the node x receives two firing messages at time t_1 and t_2 respectively. In contrast, in the RFA model node x will jump at time t_3 in the beginning of the next period as shown in Figure 2b. The sum of phase jumps can be calculated through the following equation:

$$\Delta\phi_{tot} = \sum_{i=1}^{n} \Delta\phi_i \tag{5}$$

where n is the number of firing events received in a period.

3.3 Inter-cluster Synchronicity

After all clusters achieve the synchronicity condition, every cluster can be considered as a converging node group. It can be assumed as a firing node if its phase comes near to one. In this case, the RFA mechanism can still be used to reach the synchronicity condition for all nodes in the network. The synchronicity condition can be completed through sending the firing message among the node groups. The number of node groups will decrease, which are caused by the merging of a few of the node groups into one group and finally there is only one group as the synchronicity condition of the network.

In a period the group G_x will receive the firing messages from the firing group G_y equal to the number of the G_y's members. The sum of phase jumps of group G_x is as follows:

$$\Delta\phi_{(tot),G_x} = \sum_{i=1}^{n_y} \Delta\phi_{i,G_x} \tag{6}$$

where n_y is the number of G_y's members. This enlarged the sum of phase jumps of group G_x, so that it can cause an overshooting jump in the nodes of the groups G_x. To avoid this event, the RFA model is added a Late Sensitive Window (LSW) introduced by [11], where the window inhibits the response of node against the firing messages falling outside of the window. When group G_x achieves phase $\phi = 1$ (or $t = T$), it fires and adjusts its phase based on the sum of phase jump in the previous period and jumps in the beginning of next period. The phase jump of the group G_x is calculated by the following equation:

$$\Delta\phi_{G_x} = \begin{cases} 0 & \phi \leq LSW \quad \text{and} \quad \epsilon\phi \geq 1 \\ \varepsilon\phi & \phi \geq LSW \quad \text{and} \quad \epsilon\phi \leq 1 \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

4 Simulation Results

The performance of both RFA and NCIPP-RFS method is evaluated using a network simulator NS-3 to obtain the convergence rate. A total of 40 simulations were run for each coupling strength in all scenarios. Hence, results presented in Figure 4, 5 and 6 for each coupling strength show the average values and the corresponding 95% confidence interval. The network topology uses a mesh type in a variety of sizes of 25, 50, 75, and 100 nodes spread uniformly in a constricted rectangular area sized 1000 x 1000 m^2. Furthermore, the network is designed to represent a realistic environment to evaluate the convergence rate of the synchronicity. Several simulation parameters used to evaluate the convergence rate of the synchronicity are varied in the simulation both RFA and NCIPP-RFS model as shown in Table 1. The coupling strengths (ε) are varied in the range of 0.01 through 0.1 for cluster 2, 3, 4, and 5. The metric evaluation used to measure the convergence rate is the number of periods required to reach the synchronicity condition.

Table 1: Simulation parameters

Parameter	**Value**
Oscillating period T (milliseconds)	100
Phase of oscillator ϕ	$\phi \in$ [0,1]
Phase rate $\frac{d\phi}{dt}=\frac{1}{T}$	0.01
Coupling strength ϵ	$\epsilon \in$ [0.01, 0.1]
The number of nodes	25, 50, 75, 100
The number of clusters	2, 3, 4, 5

Figure 4 shows a set of charts that evaluates the influence of the coupling strength parameter on the number of periods in both the RFA and the NCIPP-RFS. It presents some comparison of the convergence rate between the RFA and the NCIPP-RFS with various number of clusters. It can be seen that the convergence rate of the RFA is comparable to the NCIPP-RFS in 25 nodes. However, at the number of nodes, 50, 75 and 100, the convergence rate of the RFA is much slower than the NCIPP-RFS, especially in the lower coupling strength of 0.01 through 0.05. This happens because of the influence of the clustering way in the less the number of nodes to increase the convergence rate does not contribute significantly. Actually, the such way is to synchronize some groups of nodes in a network concurrently to reduce the load of the network. If node has not been synchronized in the network, all other nodes must keep the effort to synchronize it. Therefore, the clustering method of 25 nodes in the RFA results in the convergence rate similar to the NCIPP-RFS. Otherwise, in the larger the number of nodes (50, 75, and 100 nodes), the

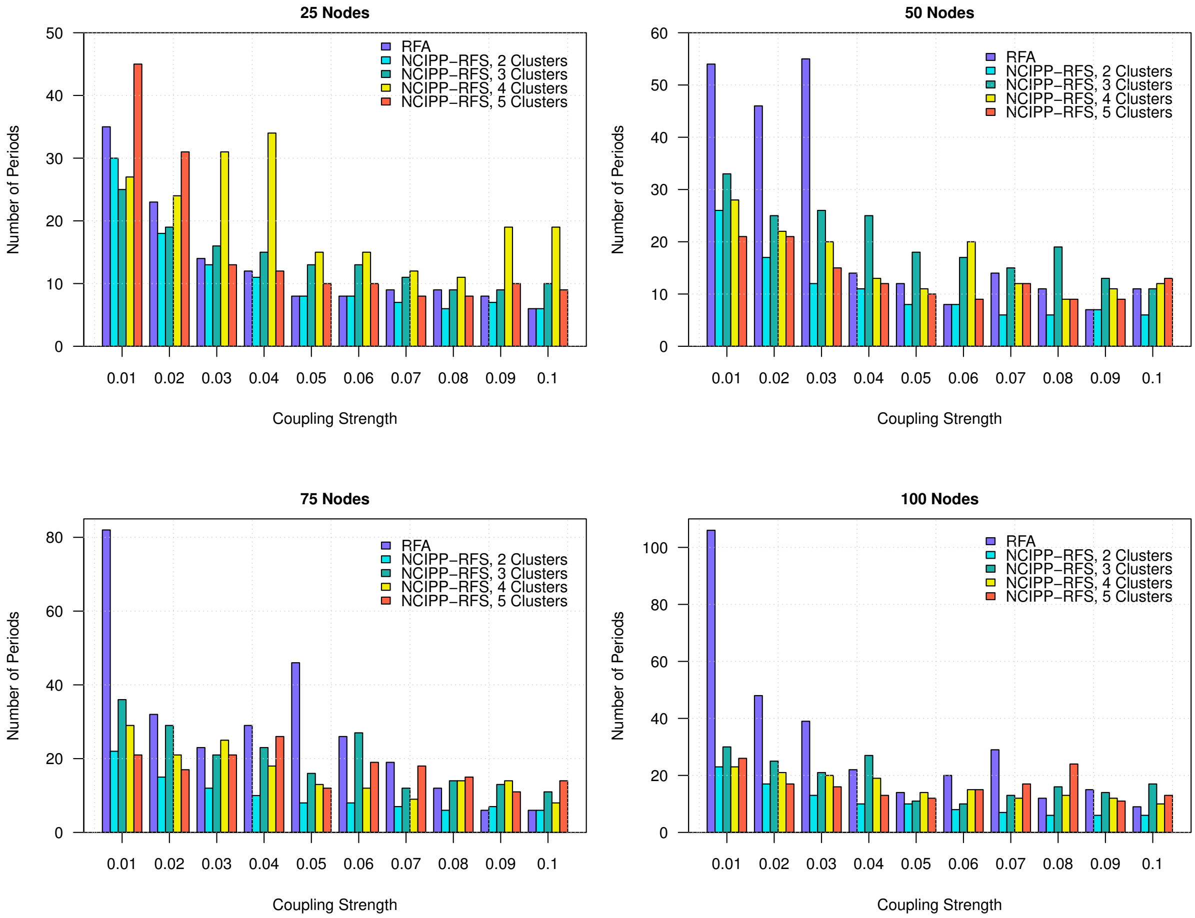

Figure 4: The comparison of the convergence rate between the RFA and the NCIPP-RFS model when the number of clusters is varied for 25, 50, 75, and 100 Nodes.

convergence rate of the NCIPP-RFS declines along with the increasing number of nodes. The influence of the clustering method in synchronicity that contributes significantly to the increase in the convergence rate is in the larger the number of nodes. Finally, the number of nodes which are greater than or equal to 50 nodes generates the best performance of the NCIPP-RFS among all scenarios when the number of node is varied.

A set of charts presented in Figure 5 is to discover the best performance among all scenarios when number of clusters is varied in the NCIPP-RFS model. The chart on the top left shows the convergence rate of the NCIPP-RFS in the two-clusters scenario. It shows that the number of periods declining rapidly along with the gradual increase of the coupling strength value in all scenarios of the number of nodes. This means that the NCIPP-RFS of two clusters will show a better performance. This is in line with the aim of the adjustment of the coupling strength parameter to increase the convergence rate gradually. Next, in the chart for the experiment with three clusters, the convergence rate slows down along with the rising number of nodes to 75 nodes, but it can increase the convergence rate in 100 nodes. Similarly, the larger the number of clusters (4 and 5) the NCIPP-RFS is slower than in the scenario for two clusters. This happens because the increase of the number of clusters causes the increase of the number of periods to reach a convergence in the inter-cluster synchronicity, so that they will slow down the convergence rate of the synchronicity process in the network. Therefore, the best performance of the NCIPP-RFS among all scenarios of the number of clusters is the scenario of two clusters.

An important problem that needs to be understood is the caution of the highly different convergence rate to the others as shown in Figure 5. They are 50 nodes with coupling strength of 0.01 in two clusters, 75 nodes with coupling strength of 0.01 in three clusters, 25 nodes with coupling strength of 0.04 in four clusters, and 25 nodes with coupling strength of 0.01 in five clusters. These problems are caused by the process of inter-cluster synchronicity that is hard to reach a synchronicity percentage of 100 percent as shown in Figure 6. It can be seen that the graphs express the movement of the synchronicity percentage that is constant in some periods causing a difficulty to rise forward to 100 percent. This problem will slow down the convergence rate of the synchronicity.

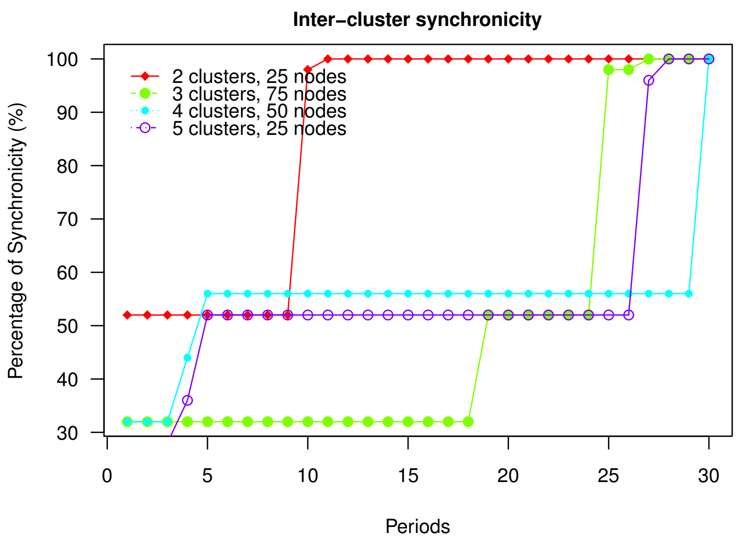

Figure 6: Inter-cluster synchronicity that is difficult to reach a synchronicity convergence

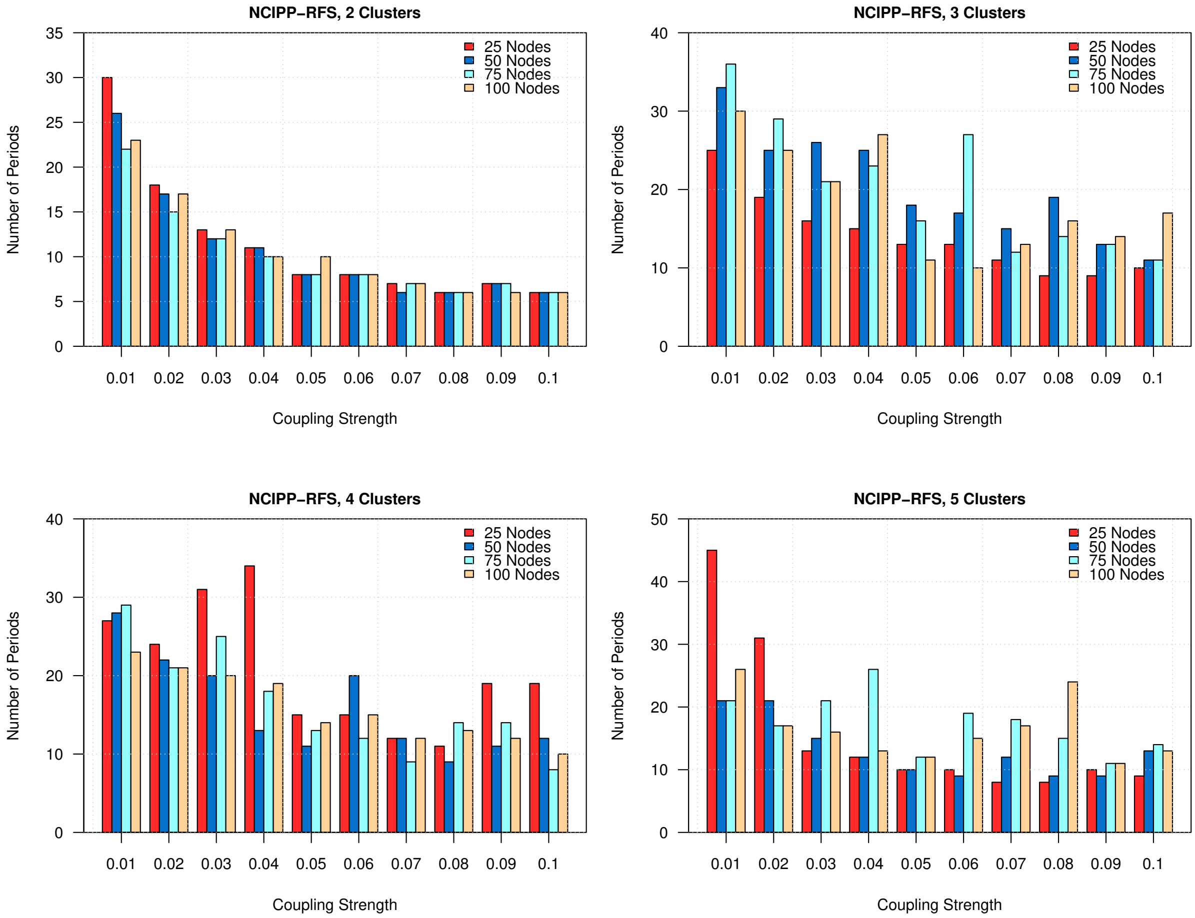

Figure 5: The comparison of the convergence rate among the number of nodes in the varied number of clusters.

Conclusions

The clustering scheme approach based on the initial phase proximity is a way to synchronize all nodes in a network executed concurrently. The clusters are constructed through a self-configured way. The intra-cluster synchronicity uses the Reachback Firefly Algorithm (RFA) simplifying the firing function and considering the realistic transmission effects. The inter-cluster synchronicity utilizes the RFA with the Late Sensitive Windows (RFA with LSW) model to avoid the overshooting jumps due to the flood of the firing messages from a node group to others.

We compare the RFA and the NCIPP-RFS clustering scheme using the Network Simulator (NS-3). The simulation results show that the convergence rate of the NCIPP-RFS can reach a synchronicity faster than the RFA. Moreover, the NCIPP-RFS can contribute significantly to the increase of the convergence rate if the number of nodes is greater than or equal to 50 nodes.

The NCIPP-RFS was also evaluated and compared in the various number of clusters. The best performance of the NCIPP-RFS among all scenarios of the number of clusters is the scenario of the least number of clusters. However, the movement of the synchronicity percentage that is constant in some periods is difficult to rise toward the 100 percent.

As future work, we will implement our approach to test it in a data-gathering application referring a temporal and spatial relationships.

Acknowledgement

The authors gratefully acknowledge the Ministry of Research, Technology and Higher Education of Indonesia for supporting this work through DIKTI's research grant under grant No. 1173/UN2.R12/HKP.05.00/2016

Bibliography

[1] F. Lamonaca, A. Gasparri, E. Garone, and D. Grimaldi (2014), Clock Synchronization in Wireless Sensor Network With Selective Convergence Rate for Event Driven Measurement Applications, *Instruments. IEEE Trans.*, 63(9): 2279-2287.

[2] F. Gielow, G. Jakllari, M. Nogueira, and A. Santos (2015), Data similarity aware dynamic node clustering in wireless sensor networks, *Ad Hoc Networks*, 24: 29-45.

[3] H. Araujo, W. L. T. de Castro, and R. Holanda Filho (2010), A proposal of self-configuration in Wireless Sensor Network for recovery of broken paths, *Sensors Applications Symposium (SAS), 2010 IEEE*, 245-250.

[4] K. Kyungmi, K. Hyunsook, and H. Youngchoi (2009), A Self Localization Scheme for Mobile Wireless Sensor Networks, *Computer Sciences and Convergence Information Technology, 2009. ICCIT 09. Fourth International Conference on*, 774-778.

[5] A. K. Mohapatra, N. Gautam, and R. L. Gibson (2013), Combined Routing and Node Replacement in Energy-Efficient Underwater Sensor Networks for Seismic Monitoring, *Oceanic Engineering, IEEE Journal of*, 38: 80-90.

[6] R. Lara, D. BenĂtez, A. Caamato, M. Zennaro; J. L.Rojo-Alvarez (2015), On Real-Time Performance Evaluation of Volcano-Monitoring Systems With Wireless Sensor Networks, *IEEE Sensors Journal*, 15(6): 3514-3523.

[7] M. M. Afsar and M.-H. Tayarani-N. (2014), Clustering in sensor networks: A literature survey, *J. Netw. Comput. Appl.*, 46: 198-226.

[8] R. E. Mirollo and S. H. Strogatz (1990), Synchronization of pulse-coupled biological oscillator, *SIAM J. Appl. Math.*, 50(6): 1645-1662.

[9] N. Yu, B. J. d'Auriol, W. Xiaoling, W. Jin, C. Jinsung, and L. Sungyoung (2008), Selective Pulse Coupling Synchronicity for Sensor Network, *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*,123-128.

[10] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal (2005), Firefly-inspired sensor network synchronicity with realistic radio effects, *Proc. of the 3rd international conference on Embedded networked sensor systems*, ACM, San Diego, California, USA, 142-153.

[11] A. Tyrrell, G. Auer, and C. Bettstetter (2006), Fireflies as Role Models for Synchronization in Ad Hoc Networks, *Bio-Inspired Models of Network, Information and Computing Systems*, 1-7.

[12] J. Yanliang, M. Wei, B. Zhishu, and Z. Xuqin (2013), Blind and buffer phase area based on M&S model fireflies synchronization in WSNs, in Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), *2013 First International Symposium on*, 1-4.

[13] C. Lin and W. HongPeng (2009), Reachback Firefly Synchronicity with Late Sensitivity Window in Wireless Sensor Networks, *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*, 1: 451-456.

[14] F. A. dos S. Silva, S. R. Lopes, and R. L. Viana (2016), Synchronization of biological clock cells with a coupling mediated by the local concentration of a diffusing substance, *Commun. Nonlinear Sci. Numer. Simul.*, 35: 37-52.

[15] Y. Yaniv, I. Ahmet, J. Liu, A. E. Lyashkov, T.-R. Guiriba, Y. Okamoto, B. D. Ziman, and E. G. Lakatta (2014), Synchronization of sinoatrial node pacemaker cell clocks and its autonomic modulation impart complexity to heart beating intervals, *Heart Rhythm*, 11(7):1210-1219. doi: 10.1016/j.hrthm.2014.03.049.

[16] D. Kim (2004), A spiking neuron model for synchronous flashing of fireflies., *Biosystems*, 76(1-3): 7-20.

[17] R. Leidenfrost and W. Elmenreich (2009), Firefly clock synchronization in an 802.15. 4 wireless network, *EURASIP J. Embed. Syst.*, 1-17, DOI: 10.1155/2009/186406.

[18] S. Yi, J. Qing, and Z. Kai (2012), A clustering scheme for Reachback Firefly Synchronicity in wireless sensor networks, *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, 27-312.