ESTIMATING THE MAXIMUM SYNCHRONIZATION DRIFT IN SELF ORGANIZING WIRELESS SENSOR NETWORKS

S. I. Pella

Electrical Engineering Department, Faculty of Science and Engineering
University of Nusa Cendana, Adi Sucipto, Kupang, 85000, Indonesia
Email: s.i.pella@staf.undana.ac.id

ABSTRACT

The nodes in self-organizing wireless sensor networks (WSNs) do not have means for synchronizing their clock with a global reference. To maintain a local synchronization, the nodes to periodically propagate a small control packet called SYNC. The propagation delay suffered by the packet caused the synchronization drift in the network, which is proportional to the length of the propagation path. This paper aims to investigate ways to calculating the maximum synchronization drift that could happen in a network. We propose a method to theoretically calculate the upper bound of the maximum schedule drift in a network, which serve as the worst case scenario. We then run simulations to see the effect of network density and node's transmission range to the maximum synchronization in the network for various network sizes. Finally, we propose a way to estimate the maximum synchronization drift in a dense network.

Keyword: *Synchronizatiod Drift, Self-OrganizingWSN, wireless sensor networks*

1. INTRODUCTION

Wireless sensor networks (WSNs) provide a way to remotely monitor physical environments via wireless channel. WSNs have been implemented in various areas of applications, such as health systems [1-3], smart home networks [4, 5] and disaster monitoring systems [6-10]. WSNs comprise a large number of densely deployed autonomous sensors called nodes monitor and gather the data of physical environment and send them to the sinks to be reported to the outside network.

Due to several reasons, nodes in some WSN applications are randomly deployed. Forest fire monitoring applications, for example, employ a large number of nodes to cover the entire forest area so that placing each node in a specific location is cost and time inefficient. Nodes are commonly deployed by throwing them out from an airplane and upon waking up need to coordinate to construct a network. Such a network is called a self-organizing network.

In most WSNs, energy conversation is among the crucial issues since nodes in WSNs are commonly battery-powered devices with limited energy resources. Duty cycle based protocols, such as S-MAC [11, 12], propose a way to conserve energy in WSN by allowing the nodes to turn off their radio modules periodically when they do not participate in packet transmission. The operational time in a node is divided into frames, where each frame consist of an active period where nodes turn their radio modules on and sleeping period where nodes turn their radio modules off. Each node maintain a schedule which determine the start of its next frame. Nodes periodically broadcast a packet called SYNC to advertise its schedule.

Nodes in WSNs need to maintain a local synchronization to communicate with their neighbor. The long synchronization drift could cause a serious error in data transmission. In a low data rate network where the bit duration is much longer than the schedule drift, this would not pose a problem. However, as the data rate of the network increases, if the bit duration is smaller than the synchronization drift, this could cause an error bit interpretation.

This study focuses on investigating the maximum synchronization drift in a randomly deployed WSNs. This value can be used in determining the important parameters in designing a network, such as the length of the guard bits, the maximum data rate allowed in the network and the maximum schedule drift in duty-cycle networks. The rest of the paper is organized as follows. Section 2 describes the synchronization issues of randomly deployed WSNs. Section 3 proposed two methods to calculate the upper bound and expected maximum synchronization drift in a network. Section 4 presents and discussed the results of the proposed methods and Section 5 summarizes the paper.

2. METHODS TO ESTIMATE SYNCHRONIZATION DRIFTS IN A NETWORK

2.1. Synchronization Drift in Long Chain WSNs

In a self-organizing WSN, nodes need to deal with synchronization issues that are slightly different from the ones in the other types of networks due to the following conditions.

(1) Each node comes to life and starts its clock at slightly different times.
(2) The nodes do not possess knowledge of their neighbourhood, including the distance to their neighbors
(3) Synchronizing the clock with an outside sever (e.g. The use of GPS satellite synchronization) in most cases is hardly an option due to (a) the cost, (b) the difficult location of the nodes, (c) the limited energy the nodes have.

There are two types of possible synchronization problems with this type of network.

(1) Synchronization drift due to the inaccuracy of internal clocks in the nodes.
(2) Synchronization drift due to the propagation delay of the SYNCs. Propagation delay in the network causes nodes to see the start (and consequently the end) of a frame slightly differently according to their position in the network.

Numerous studies, such as [13-17] have proposed algorithms for clock synchronization in WSNs, therefore the synchronization drift discussed in this study is focused on the drift caused by propagation delay in the network.

As proposed in S-MAC, nodes in WSNs maintain local synchronization by periodically broadcasting *SYNCs*. The reception nodes receive the packets after some amount of delay, including processing delay, transmission delay, and propagation delay. These delays make different receivers 'see' the start of the advertised frame differently. While the receivers can easily correct the drifts due to the transmission and processing, there is no way to calculate the drift due to the propagation delays without knowing the length of propagation path the packet travelled.

We consider the case of a network with four nodes that are connected in a chain topology as shown in Figure 1. Because of the propagation delays of the SYNC packets, nodes *B*, *C* and *D* 'see' the start of the schedule slightly behind the time that node *A* 'sees' it . Assuming that the SYNC packets have an average propagation delay of d, if node *A* advertises schedule S starts at time t then nodes *B*, *C*, and *D* see the schedule start at time $t+d$, $t+2d$, and $t+3d$ respectively.

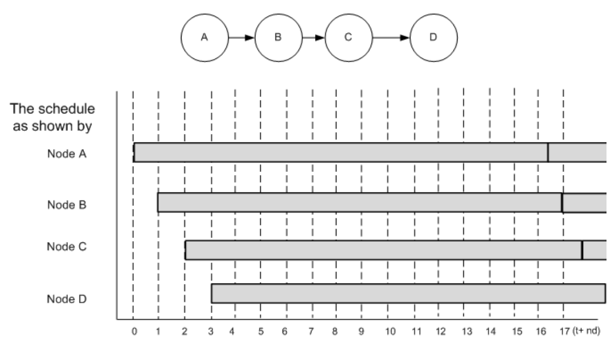

Figure 1 Schedule Drift in a Chain Network

The drift caused by a propagation delay is generally small and easy to correct using the guard bits. However, in a long chain network, the accumulated drift could cause a problem. We consider a long chain network with $n+1$ nodes where node $N_{(n+1)}$ is in the transmission range of node N_1 and node N_n as shown in Figure 2.

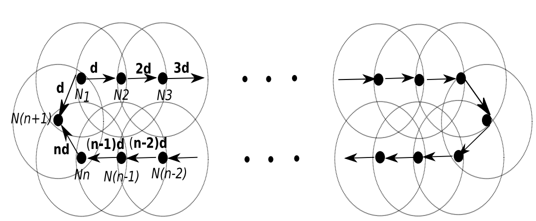

Figure 2 Schedule Drift in a Long Chain Network

In Figure 2, the nodes $N_1, N_2, .. N_n$ are connected in a chain topology and implement a common schedule S. Because of the propagation delays of the SYNCs, *Nn* starts its frame *(n-1)d* time unit after N_1 does. Consider a case where new node $N_{(n+1)}$ recently joins the network and start its discovery period. During the discovery period, it receives two SYNCs from N_1 and N_n announcing schedule *S*. If, for example, the SYNC sent by N_1 announces that the next frame starts at time *t*, then, consequently, the SYNC sent by N_n - announces that the next frame starts at time *t+(n-1)d*. There are two issues that *N(n+1)* needs to address. Firstly, N_{n+1} needs to realize that the SYNC packets of N_1 and N_n actually announce the same schedule. Secondly, for example, if it decides to synchronize itself with node N_1, it then receives the packets from node *Nn* with *(n-1)d* unit time drift.

The synchronization drift between any two nodes in a network is given by

$$\delta = \frac{L}{c} \qquad (1)$$

Where L is the length of the propagation path between the two nodes and c is the speed of light.

In the next section, we discuss methods to determine the maximum synchronization network drift in a network by estimating the maximum propagation path length, L_{max}, of the network.

2.2. THE THEORETICAL UPPER BOUND OF SYNCHRONIZATION DRIFT IN A NETWORK

This section discusses the upper bound of schedule drift in a network which serves as the worst case scenario.

Proposition 1 *Let the network be a square network with a dimension of M x M and nodes with a uniform transmission range (r) are placed in particular positions in the network to create the maximum propagation path length given the area of the network, as shown in Figure 3. Then the maximum synchronization drift in the network is given by*

$$\delta_{\max\_upper} = \frac{2}{\sqrt{3}}\frac{A}{cr} \qquad (2)$$

*where **A** is the area of the network.*

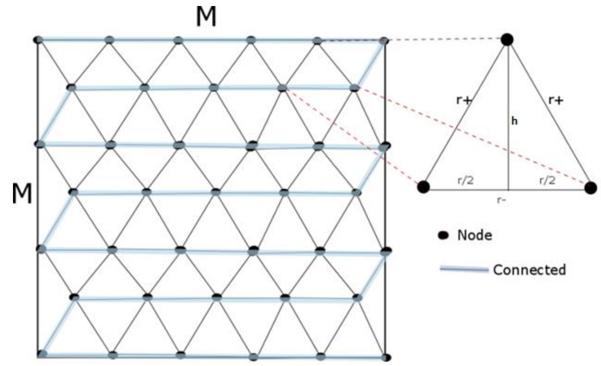

Figure 3 Upper Bound of Longest Chain in a Square Network

Proof.

(1) In Figure 3, the maximum distance between two connected nodes is *r-* (slightly less than *r*) and the minimum distance between two nodes that are not connected is *r+* (slightly larger than *r*).

(2) Because of the geographical span, the chain needs to be ‘folded back’ to fit the dimension of the network. The distance between folds is given by $h = \frac{1}{2}r\sqrt{3}$.

(3) The maximum number of ‘folds’ in the network, $F = \frac{M}{h} = \frac{2M}{r\sqrt{3}}$.

(4) The number of hops in each ‘fold’, $H = \frac{M}{r}$.

(5) Finally, we can calculate the maximum number of hops in a network, *N*, as follows

$$N = FH = \frac{2M^2}{r^2\sqrt{3}}$$

(6) Assuming that a packet in the network propagates at the speed of light, we have the maximum synchronization drift as the maximum physical distance the SYNC packet propagated through divided by the speed of light

$$\delta_{upper} = \frac{L_{max}}{c} = \frac{Nr}{c} = \frac{2}{\sqrt{3}}\frac{M^2}{cr}$$

(7) Noting that M^2 is the physical area of the network, we can derive the upper bound of schedule drift in a network with an area of A as shown in equation (2).

$$\delta_{upper} = \frac{2}{\sqrt{3}}\frac{A}{cr}$$

2.3. Expected Value of Maximum Synchronization drift in a Dense Network

2.3.1 Density and Connectivity of randomly deployed network

Given a specific network area, the number of nodes deployed in the network determines the density of the network. Figure 4 shows an example of the connectivity in randomly deployed networks with a fixed network size (5r x 5r, r = transmission range) and various densities (density= 1,2,3,5). In a very sparse network, as shown in Figure 2(i), the connectivity is so low that the network is partitioned into smaller isolated networks. As the density of the network increases, so does the connectivity of the network (ii). In (iii), the network is already fully connected. From this point, the connectivity of the network remains while the density of the network continues to increase and causes more nodes to be available in a position that can create a shorter path of the two furthest nodes in a network (iv).

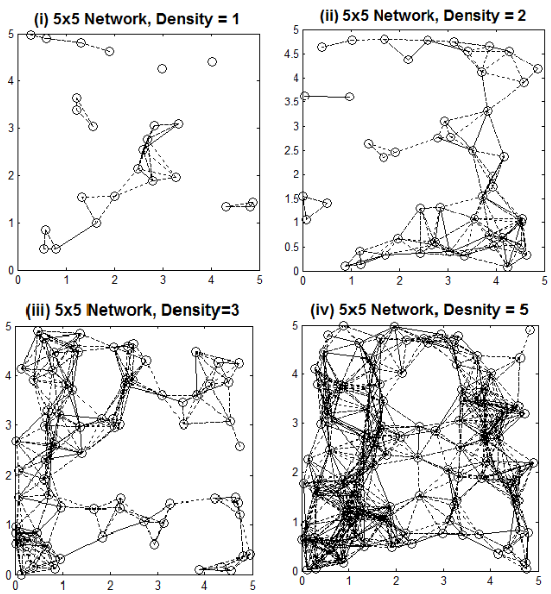

Figure 4 Network with Various Densities

The relation between the connectivity of a network and the network density for various sizes of networks is shown in Figure 5. The solid lines on the graph show the mean of the network connectivity in 100 simulations and the dashed lines on the graph show the percentage of times the networks reach 100% connectivity (fully connected) in 100 simulations.

Figure 6 shows the number of hops between the two furthest nodes in a uniformly deployed network of various network sizes and densities. The solid lines show the average values of 100 simulations and the dashed lines show the maximum values of 100 simulations. A low density network forms a low chain length due to isolated groups of nodes. As the density increases, so does the chain length since more nodes are connected (as shown in Figure 5). It is important to note that the chain length in the simulation is much lower than the upper bound chain length (N) in section 3.3. In the next subsection, we provide a way to estimate the chain length and the synchronization drift between two farthest nodes in the network.

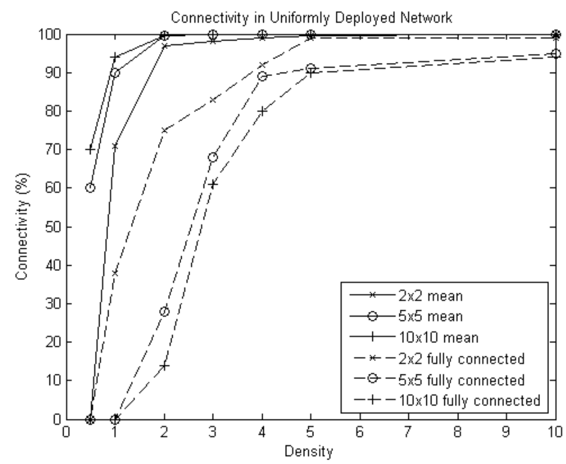

Figure 5 The connectivity in a Uniformly Distributed Network

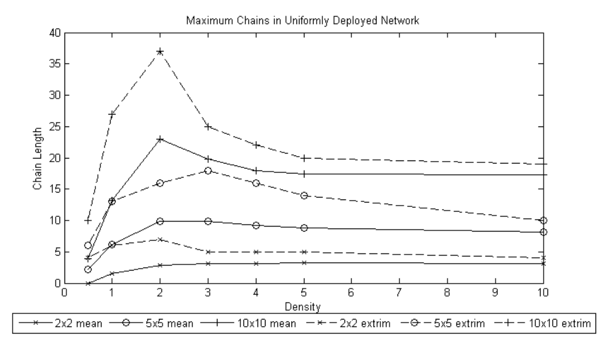

Figure 6 Chain Length in a Randomly Deployed Network

2.3.2 Estimating the Expected Maximum Synchronization Drift in a Dense Network

Section 3.1. provides an asymptotic analysis of the drift in a network where the nodes are placed specifically to create the maximum chain length. In a reality of a randomly deployed network, the probability of this condition to exist is very small and the maximum drift is supposedly considerably smaller.

This section proposes a method to calculate an expected maximum synchronization drift in a network.

Proposition 2 *Let the network be a square network with dimension MxM. As the density of the network increases the expected maximum synchronization drift in the network approach*

$$E[\delta_{max}] = \frac{M\sqrt{2}}{c} \tag{3}$$

Proof

(1) The synchronization drift between any two nodes N_1 and N_n equals to the propagation delay between node N_1 and node N_n. Assuming that packets are propagated from N_1 to N_n trough path {$N_1, N_2, \ldots, N_{n-1}, N_n$), then the synchronization drift is given by $\delta = \frac{\sum_{i=1}^{n-1} r_{i,i+1}}{c}$, where $r_{i,i+1}$ is the physical distance N_i (current node) and N_{i+1} (next hop in the propagation path).

(2) In a very dense network (e.g. Figure 2(iv)), there will be nodes positioned along the Euclidian of any two nodes in such a way that the maximum propagation path between the two nodes equal to the Euclidian's distance of the two nodes.

(3) In a square MxM network, the Euclidian of two furthest nodes in the network is less or equal to the diameter of the network, $D = M\sqrt{2}$

(4) Therefore the maximum propagation delay between the two farthest nodes is given by

$$E[\delta_{max}] = \frac{\sum_{i=1}^{n} r_{i,i+1}}{c} = \frac{M\sqrt{2}}{c}$$

3. RESULT AND DISCUSSION

3.1. Upper Bound of Synchronization Drift in a Network

In **Error! Reference source not found.**, we plot the upper bound of synchronization drift in a square network for various transmission ranges {R=1, 2, 3, 4, 5, 6, 7, 8, 9 ,10 (m)} and network sizes {2x2, 5x5, 10x10 m^2}.

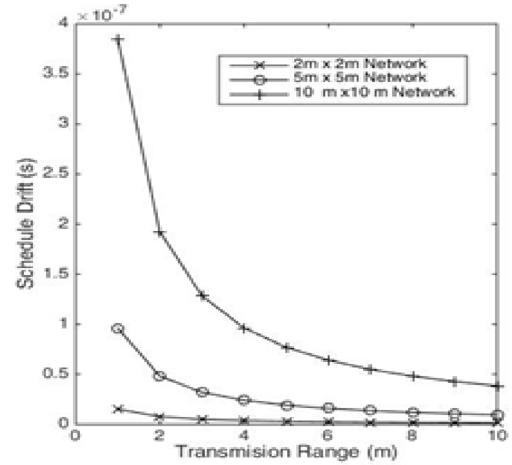

Figure 7 Upper Bound of Synchronization Drift of a Network

The result shows that the synchronization drift in the network is inversely proportional to the transmission range and proportional to the area of the network. As mentioned in section 3.1, these values served as the worst case of schedule drift in the network.

3.2. Expected Maximum Synchronization Drift in a Randomly Deployed Network

To observe a realistic maximum synchronization drift in a network where the nodes are randomly deployed, we conduct a Monte Carlo simulation in the MATLAB 2014 environment, with various network sizes (2x2, 5x5, 10x10 m^2) and densities for a unit transmission range (r=1 m). To achieve good accuracy, we run 100 simulations for each condition.

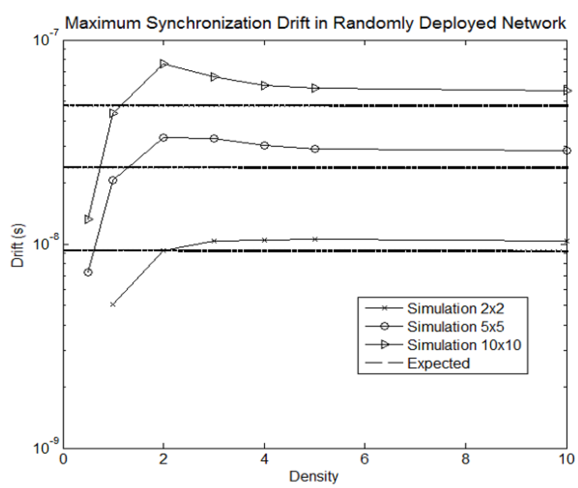

Figure 8 The Maximum and Average Value of Maximum Number of Hops in a Uniformly Deployed Network

The result shows that the maximum schedule drift in a network increase as the density increases until reaching the maximum connectivity in the network. After that, as the density increases more nodes are available near the Euclidean of two furthest nodes in the network, thus decreases the propagation path length and further the synchronization drift in the network.

4. SUMMARY

Self-organizing WSNs suffers synchronization drift due to the propagation delay of the SYNC packet which could affect the network performance. Estimating the maximum synchronization drift is important in designing key parameters of a network. This paper has proposed two ways in estimating the maximum synchronization drift. The theoretical upper bound of maximum synchronization drift gives the worst case scenario which could occur in a network with a specific physical size and transmission range. We also provide a way that gives a more realistic estimation of the maximum synchronization drift in a dense network given a specific physical area. A series of simulation shows that, for various sizes of networks, as the density increase the maximum synchronization drift approaches the estimated value.

5. REFERENCES

[1] M. Nabi, M. Blagojevic, M. Geilen, and T. Basten, "Dynamic data prioritization for quality-of-service differentiation in heterogeneous wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, 2011, pp. 296-304.

[2] A. Milenković, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer communications,* vol. 29, pp. 2521-2533, 2006.

[3] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, *et al.*, "Health monitoring of civil infrastructures using wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, 2007, pp. 254-263.

[4] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *Sensors Journal, IEEE,* vol. 12, pp. 1965-1972, 2012.

[5] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *Sensors Journal, IEEE,* vol. 13, pp. 3846-3853, 2013.

[6] A. Khadivi, L. Georgopoulos, and M. Hasler, "Forest fire localization using distributed algorithms in wireless sensor networks," in *Proceedings of PIERS*, 2009, pp. 452-455.

[7] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *sensors,* vol. 9, pp. 8722-8747, 2009.

[8] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, "A framework for use of wireless sensor networks in forest fire detection and monitoring," *Computers, Environment and Urban Systems,* vol. 36, pp. 614-625, 2012.

[9] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, *et al.*, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE,* vol. 10, pp. 18-25, 2006.

[10] N. Meenakshi and P. Rodrigues, "Tsunami Detection and forewarning system using Wireless Sensor Network-A Survey," 2014.

[11] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 1567-1576.

[12] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *Networking, IEEE/ACM Transactions on,* vol. 12, pp. 493-506, 2004.

[13] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review,* vol. 36, pp. 147-163, 2002.

[14] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 138-149.

[15] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 39-49.

[16] K. S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, pp. 244-253, 2014.

[17] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 13, pp. 384-397, 2005.