

## Design of Time Synchronization Protocol based on Master-Slave Topology for Heterogeneous USN

Yongwan Ju<sup>1</sup>, Woojin Paik<sup>2</sup> and Moonshin Shin<sup>2\*</sup>

<sup>1</sup>*Div. of Industry Development, Korea Internet & Security Agency*

<sup>2</sup>*Dept. Of Computer Engineering, Konkuk University, Korea*

*ywjju@kisa.or.kr, {wjpaik,msshin}@kku.ac.kr, \* Corresponding Author*

### Abstract

*In this paper, we propose a specific time synchronization protocol called MSTSP (Master Slave Time Synchronization Protocol). We designed MSTSP based on the master-slave topology in order to synchronize the time of the heterogeneous sensor nodes. Heterogeneous sensor networks can be used in various risk-associated areas. Time synchronization among the heterogeneous sensor nodes is essential in Ubiquitous Sensor Network systems. In MSTSP, the time-sync master nodes can always be synchronized with the Sink Node of USN, because the sensor nodes which are homogeneous with the Sink Node become time-sync master nodes. And other sensor nodes which differ from the Sink Node must be time-sync slave nodes, which need to belong to one of the specific master nodes. The time-sync master nodes then enable time-sync slave nodes to be synchronized with the sleep periods. The proposed MSTSP is also useful for power saving by maintaining maximum sleep time.*

**Keywords:** *Time synchronization, Heterogeneous sensor nodes, Sink node, Time-synch Master nodes, Time-synch Slave nodes*

### 1. Introduction

Sensor networks are wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Ubiquitous sensor networks facilitate the monitoring and controlling of physical environments from remote locations with better accuracy. Recently, improvement of the technology has made possible the deployment of small, inexpensive, low-power sensor nodes. Each sensor node is capable of not only utilizing a minimal amount of processing but also a minimal amount of power.

Thus, unlike traditional networks, the major consideration in a sensor network is to extend longevity and robustness of the system. Above all, two factors need to be seriously considered to maintain the ubiquitous sensor networks. First, energy efficiency is a dominant consideration because sensor nodes only have a small and finite source of energy. Therefore, optimizing energy usage is essential to design USN. Secondly, time synchronization is significantly referred to as a localization of USN, because in most cases, sensor nodes are deployed in an ad-hoc manner. The nodes are responsible for identifying themselves in some spatial co-ordinate system. Sensor network consists of various sensor nodes for observation, information processing, and communication functions. An ad-hoc network is automatically constructed and application services can be provided through collecting and processing necessary information after installing sensors to the service area.

In this paper, we propose a master-slave topology based time synchronization for a USN that consists of various sensor nodes including heterogeneous sensor nodes. The

proposed MSTSP (Master-Slave based Time Synchronization Protocol) can be applied to the heterogeneous USN, which consists of various sensor nodes, some of which are homogeneous with the Sink Node, while other sensor nodes are heterogeneous. The sensor nodes which are homogeneous with the Sink Node of USN are named time-sync master nodes and other sensor nodes which are heterogeneous with the Sink Node are named the time-sync slave nodes. The time-sync master nodes are available for time synchronization with the Sink Node and also enable the time-sync slave nodes to be synchronized. This means that the heterogeneous sensor nodes become time-sync slaves and are controlled by the specific time-sync master. The proposed MSTSP can be useful in the environment of heterogeneous USN and also effects a saving of power. The remainder of the paper is organized as follows. Section 2 describes related works and Section 3 presents the idea of applying master-slave topology based time synchronization for heterogeneous USN in detail. The MSTSP algorithm is also described. Section 4 explains the experimental results and Section 5 provides a brief conclusion.

## 2. Related Work

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. Sensor network protocols and algorithms must possess self-organizing capabilities. Another feature of sensor networks is the cooperative effort of sensor nodes. Sensor network applications require wireless ad hoc networking techniques [1]. Recent advances in micro-electro-mechanical systems technology, wireless communications, and digital electronics have enabled ubiquitous sensor networks. Although there have been severe resource constraints in USN, it is essential to reevaluate the traditional distributed algorithms. Time synchronization is one of the basic middleware services of the USN. In the internet domain, the most widely used protocol is the Network Time Protocol (NTP) devised by Mills [13]. However, NTP, which has been widely deployed and proved to be effective in the internet domain, was non-deterministic in transmission time due to the Media Access Channel (MAC) layer. Therefore, for further adaptation to the WSN, the Reference Broadcast Synchronization (RBS) algorithm [14] and the Timing-sync Protocol for Sensor Networks (TPSN) were introduced [14]. The TPSN achieves better performance than RBS, but it does have clock drift limitation of accuracy and does not handle dynamic topology changes. Finally, the implicit dynamic topology management of the Flooding Time Synchronization Protocol (FTSP) was introduced in [16]. FTSP provides rapid convergence and robustness. However, it does not cover the heterogeneity of the sensor nodes of the USN.

Time synchronization is important for consistent distributed control and sensing. Furthermore, services such as coordination, communication and power management also depend on the existence of a global time of USN. When the tiny sensor nodes (which consist of sensing, data processing, communicate components) are deployed once, the sensor networks are usually unattended, so it is not easy to replace battery of sensor. The need for energy-efficiency and other constraints is not found in conventional distributed systems. The energy-efficiency is very important in USN. Also, in USN, it is essential that all nodes are able to wake up at the same time to be able to exchange information. So far, various time divisions multiple access (TDMA) schemes proposed in the ad-hoc networks assume clock synchronization of the nodes [1]. A number of works have attempted to improve energy efficiency by frequently switching sensor nodes or components thereof into power saving sleep modes [17]. Although the work of [6] is suitable for wireless sensor networks synchronization, it does not cover the heterogeneity and low power management.

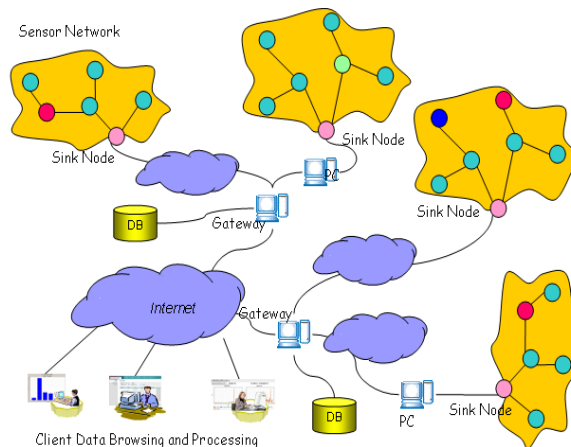
Zigbee topology provides scalable coordination of sensor nodes by a pan coordinator. Zigbee consists of three topologies: star, mesh and cluster tree. It covers time synchronization of various devices which have different clock sources according to the classification of PAN Coordinator, Full Function Device and Reduced Function Device. Pan Coordinator periodically sends a beacon message that includes information of active time and time slots. It can work for synchronization in the beacon frame. However, Zigbee topology is limited to only 5 minutes duration. Pan Coordinator must always be in the 'power on' mode in order to repeatedly send beacon messages every 5 minutes. This causes inefficient power use and it means that the longevity of energy and flexibility of time synchronization according to the applications domain cannot be guaranteed.

To solve the above problems, in this study we propose a Master-Slave Time Synchronization Protocol (MSTSP) for the heterogeneous sensor nodes in USN. When we designed the MSTSP, our goal was to achieve time synchronization in the complicated environment of USN deployed sensor nodes that previously consisted of both homogeneous and heterogeneous sensor nodes.

We implemented two phase time synch algorithms: extended FTSP for the homogenous sensor nodes and MSTSP for the heterogeneous sensor nodes. The aim of the proposed MSTSP is to provide not only time synchronization for heterogeneous sensor nodes but also longevity of energy because it is able to switch off during the maximum sleep periods. In addition, our approach is expected to be flexible with sleep periods according to the applications domain and is available for application to any kind of multi hop USN.

### 3. Master-Slave Time Synchronization Protocol for Heterogeneous USN

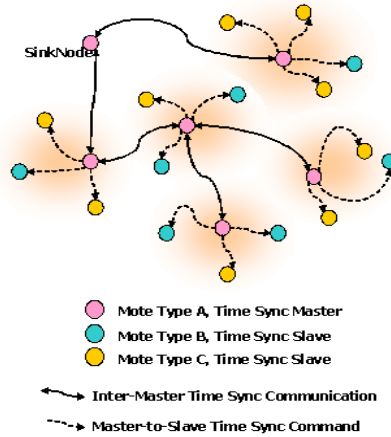
In this section, we describe the time synchronization protocol in order to provide a mechanism of synchronizing the local clocks of the sensor nodes in the heterogeneous USN.



**Figure 1. Heterogeneous USN.**

Figure 1 shows a framework for the heterogeneous USN. A USN application system is limited in power. Therefore, minimizing power use must be achieved by periodically repeating the active cycle and sleep cycle. Time synchronization enables all sensor nodes to synchronize active time and sleep cycle with accuracy. It is impossible for the FTSP algorithm to synchronize the heterogeneous sensor nodes because they have different crystal

clocks. Therefore, in a USN that includes heterogeneous motes, the existing algorithm needs to be redesigned to synchronize the active/sleep time. To solve this problem, we present the Master-Slave Topology to synchronize the active/sleep time of the heterogeneous motes.



**Figure 2. Master-Slave Topology for heterogeneous USN**

Figure 2 shows a framework for the heterogeneous USN applying the Master-Slave topology. Following this topology, there are two kinds of sensor nodes: nodes that have the same crystal clock as that of the Sink Node of the USN, and nodes that have a different clock source to that of the Sink Node.

The homogeneous mote with the Sink Node is called the Time Sync Master Node. The Time Sync Master Node plays the role of cluster header and is able to synchronize Slave-Nodes. Sink Node and Time Sync Master Node do have appropriate capabilities and performance to maintain the cluster of Slave-Nodes. Inter-Master Time Sync Communication can successfully be performed within the Sink Node and Time Sync Master Nodes.

Time Sync Slave Node has a different crystal clock than that of the Time Sync Master. So Time Sync Slave Node needs to always be under the control of specific Time Sync Master Node while changing active/sleep mode. The Slave Node must be deployed within the radio transmission range of at least one Master Node. If there are many Master Nodes, the Slave Node can select one Master Node as its parent node and be controlled by the selected Master Node.

**Table 1. Margin specifications**

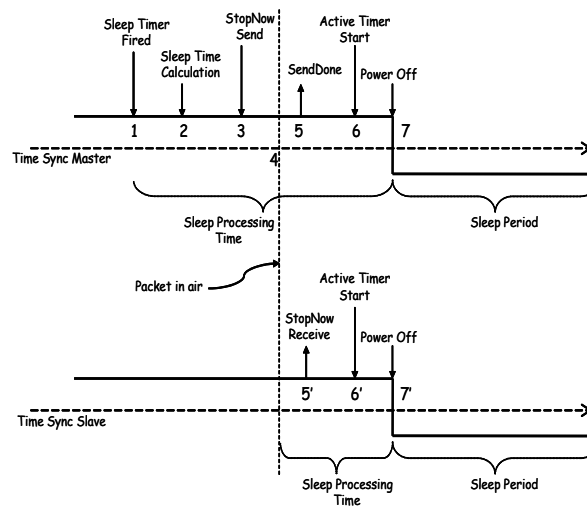
Functions	Homogeneous	Heterogeneous	
		Master	Slave
Time Sync. Protocol	Send/Receive	Send/Receive	X
Time Sync. Command	X	Send	Receive
Routing Protocol	Send/Receive	Send/Receive	Send/Receive
Parent Enable ?	O	O	X

Therefore, the Slave Node can never be a parent node of any other nodes during the network routing. The following table describes the functions of the Time Sync Master and

### Time Sync Slave Nodes.

In the case of USN with only homogeneous sensor nodes, the Time Sync Command is not required. In addition, it is not necessary to set the constraint of the enabling parent node. However, in the USN that includes the heterogeneous sensor nodes, two-phase synchronization is needed. Both the Inter-Master Time Sync Communication and Master-Slave Time Sync Command are required for the synchronization of the global time. The Master Node sends messages of the command that controls the active/ sleep modes of the Slave Nodes. The Master Node could be the parent node with respect to the network routing. Slave Nodes do not need to achieve the Time Sync. Algorithms for themselves but do receive commands from their Master Node. Then the Slave Nodes calculate the sleep time and can change the mode of repeated active or sleep time. Also, a Slave Node can never be a parent node. Master Nodes play the vital role of the header of the cluster of Slave Nodes. Our approach is flexible with sleep periods so that it is available for application to any USN applications domain.

Figure 3 shows the detail process of performing time synchronization between the Master and Slave Nodes.



**Figure 3. Control Process of Master-Slave Time Sync Command**

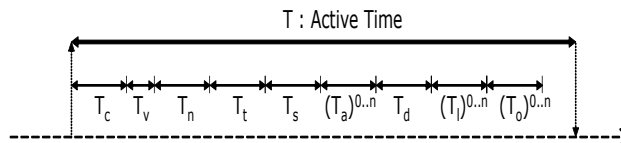
The command from Master to Slave controls the active/sleep mode of the Slave Nodes. Before it becomes the sleep mode, the Master Node calculates the global time and broadcasts the ‘Stop Now’ message, which includes the information of the time stamp for the Slave Node about how long to be asleep. As soon as the Slave Node receives the ‘Stop Now’ message from the Master Node, it changes the mode to sleep.

Each step is described in detail as follows.

- **[Master] Sleep Timer Fired:** Sensor node enters the active mode and is set to sleep time in the timer. The software module is then interrupted by the timer when the sleep time has been fired. The Master node is then changed to sleep mode.
- **[Master] Sleep Time Calculation:** To satisfy the sleep time defined by the applications, the Master node needs to calculate its own sleep time. As each node is able to have a different sleep time for each active/sleep cycle, sleep time calculation is repeatedly performed every cycle. The calculated time is  $T_s$ .

- **[Master] “Stop Now” Send:** By using the radio chip, the Master node achieves the control packet of the Slave node, which is called “Stop Now” send. The packet includes the information of  $T_s$ .
- **[Packet in air]:** While the Master node broadcasts the packet, all nodes near the Master node will also receive the packet
- **[Master] Send done:** After a radio chip finishes a stage of sending the data, the radio chip makes ‘interrupt’ of send done to the software module.
- **[Slave] “Stop Now” Receive:** After a radio chip receives the data and the radio chip makes ‘interrupt’ of stop now to the software module.
- **[Master] Active Timer Start:** During this period,  $T_s$ , the calculated sleep time, is set to the active timer in order to make interrupt of firing sleep time.
- **[Slave] Active Timer Start:** During this period, the active timer is set to be  $T_s$ , which was sent from the Master
- **[Master] Power Off & 7’. [Slave] Power Off:** During the sleep mode, most components of the sensor nodes such as Sensor H/W, Radio chip, and Clocks, except the clock of the fired active timer, are set to ‘power off’ in order to save energy.

The task scheduling is first designed in order to implement the proposed MSTSP. It is a sequence of the tasks that are performed by all sensor nodes during active mode. All sensor nodes start in the active mode at the same time and task scheduling is essential for the accuracy of time synchronization of the entire USN to avoid interference with adjacent nodes in the application domain. Figure 4 shows the task scheduling for all nodes during the active mode. All summation times must be less than the active time.



**Figure 4. Task Scheduling for MSTSP**

Details are as shown in the following.

- $T_c$  : Command processing

$T_c$  is the task for processing the command from gateway to sensor node. It can be a solution for the miscellaneous differences of time caused by time synchronization. Therefore it is the first task.

- $T_v$  : Variable Time

$T_v$  is the random time for each sensor node to do nothing. It provides a different time for each task and prevents both the competition of the channel and any failure in sending the message.

·*T<sub>n</sub>* : Network Routing Time

*T<sub>n</sub>* is the time for sending the network routing message. Sensor node receives the message sent.

·*T<sub>t</sub>* : Time Sync. Time

*T<sub>t</sub>* is the time for processing the time synch message to be sent by Master Node. In the case of Time Sync Slave, the Slave Node never performs this task.

·*T<sub>s 0..n</sub>* : Sensing Time

Sensing task is performed during *T<sub>s 0..n</sub>*.

·(*T<sub>a</sub>*)*0..n* : Actuator Time

Actuator is actuated during (*T<sub>a</sub>*)*0..n*. Through the actuator, the environment of sensor nodes is able to be controlled. If the actuator is not needed, *T<sub>a</sub>* is omitted.

·(*T<sub>d</sub>*) : Sensing Data Sending Time

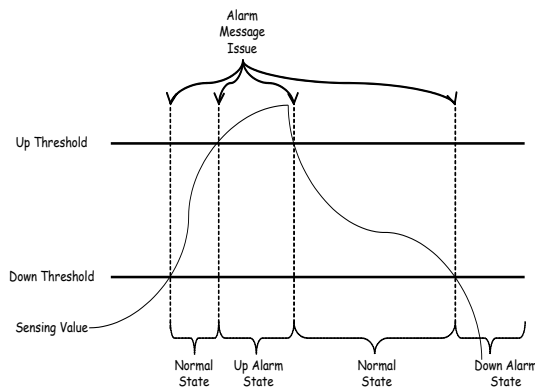
During this time, the sensing data collected by *T<sub>d</sub>* is sent to the Sync Node of USN.

·(*T<sub>l</sub>*) *0..n* : Alarm Message Sending Time

When the alarm is occurred, the task of sending the alarm message to Sync Node is achieved.

·*T<sub>o</sub>* : Operation Information Sending Time

A command to request operation information from gateway is sent and the task of sending the message to Sync Node is then performed. Operation information includes the operations of the sensor node such as active/sleep time, radio power, and the threshold of each sensor.



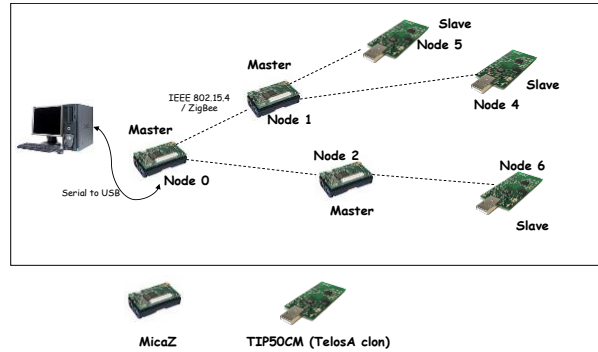
**Figure 5. Sensing data filtering using Up/Down threshold**

Although many tasks were performed, all messages are to be processed normally within the active time. While all sensing data must be sent to Sink Node or gateway, the collected data has a large volume, so the sensing data needs to be filtered before sending to gateway.

A simple filtering is used for alarm management. We use the down/up thresholds not only to select sensing data but also to check the anomaly. The users can define the UpThreshold and DownThreshold. If the sensing value did not include these ranges, anomaly alarms could be made. Figure 5 shows the states of the alarms and thresholds.

## 4. Experiments

In this section, we describe the experiments to evaluate the time synchronization of heterogeneous sensor nodes applying master-slave topology.



**Figure 6. Heterogeneous virtual USN for experiments**

We constructed the simulated environment of the virtual USN including both homogeneous sensor nodes and heterogeneous sensor nodes. The proposed MSTSP was implemented on the MicaZ as Time-Sync Master Node and TIP50CM (TelosA clon) as Time-Sync Slave Node. We observed that the time sensing data had arrived at the gateway and verified the synchronized time of the heterogeneous sensor nodes.

**Table 2. Hardware platform for simulated environment**

	Master	Slave	
Name	MicaZ	TIP50CM (TelosA clon)	
Processor	8bit RISC 16MHz	16bit RISC 8MHz	
Radio Chip	CC2420	CC2420	IEEE 802.15.4 2400 MHz ~ 2483.5 MHz
Clock	921.6 khz	32khz	
Range	10m ~ 100m	10m ~ 100m	
Sensor	Temporature, Photo	Temporature, Photo, Humidity	
Battery	AA 2EA	AA 2EA	
Power	2.7 ~ 3.3 V	2.7 ~ 3.3 V	

We implemented MSTSP and evaluated experiments. MSTSP performs time synchronization in two stages. In the first stage, multi-hop routing is performed. Time synchronization of heterogeneous sensor nodes is carried out in the second stage. Figure 7 shows that implemented routing algorithm works well for the heterogeneous USN.



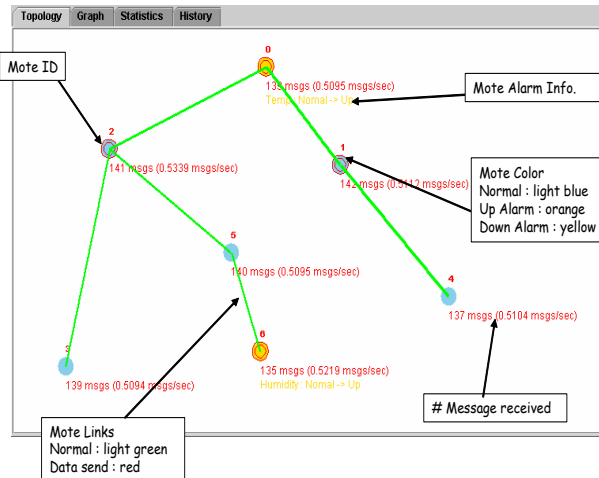


Figure 7. Results of update routing

Using the arrival times of the sensing data to the gateway ensures that the master nodes coordinate the slave nodes to synchronize the sleep cycle by calculating the sleep time only when the master nodes are invoked.

Count	Node 1		Node 2		Node 4		Node 5		Node 6	
	Time	Cycle Num	Time	Cycle Num	Time	Cycle Num	Time	Cycle Num	Time	Cycle Num
1	00:00:30	704	00:00:30	719	00:00:30	584	00:00:30	684	00:00:30	665
2	00:01:05	705	00:01:05	720	00:01:05	585	00:01:05	685	00:01:05	666
3	00:01:40	706	00:01:40	721	00:01:40	586	00:01:40	686	00:01:40	667
4	00:02:14	707	00:02:14	722	00:02:15	587	00:02:15	687	00:02:15	668
5	00:02:49	708	00:02:49	723	00:02:49	588	00:02:49	688	00:02:49	669
6	00:03:24	709	00:03:24	724	00:03:24	589	00:03:24	689	00:03:24	670
7	00:03:59	710	00:03:59	725	00:03:59	590	00:03:59	690	00:03:59	671
8	00:04:34	711	00:04:33	726	00:04:34	591	00:04:34	691	00:04:34	672
9	00:05:08	712	00:05:08	727	00:05:08	592	00:05:09	692	00:05:09	673
10	00:05:43	713	00:05:43	728	00:05:43	593	00:05:43	693	00:05:43	674
11	00:06:18	714	00:06:18	729	00:06:18	594	00:06:18	694	00:06:18	675
12	00:06:53	715	00:06:53	730	00:06:53	595	00:06:53	695	00:06:53	676
13	00:07:28	716	00:07:27	731	00:07:28	596	00:07:28	696	00:07:28	677
14	00:08:02	717	00:08:02	732	00:08:02	597	00:08:03	697	00:08:03	678
15	00:08:37	718	00:08:37	733	00:08:37	598	00:08:37	698	00:08:37	679
16	00:09:12	719	00:09:12	734	00:09:12	599	00:09:12	699	00:09:12	680
17	00:09:47	720	00:09:47	735	00:09:47	600	00:09:47	700	00:09:47	681
18	00:10:21	721	00:10:21	736	00:10:22	601	00:10:22	701	00:10:22	682
19	00:10:56	722	00:10:56	737	00:10:56	602	00:10:56	702	00:10:56	683
20	00:11:31	723	00:11:31	738	00:11:31	603	00:11:31	703	00:11:31	684
21	00:12:06	724	00:12:06	739	00:12:06	604	00:12:06	704	00:12:06	685
22	00:12:41	725	00:12:40	740	00:12:41	605	00:12:41	705	00:12:41	686
23	00:13:15	726	00:13:15	741	00:13:15	606	00:13:16	706	00:13:16	687
24	00:13:50	727	00:13:50	742	00:13:50	607	00:13:50	707	00:13:50	688
25	00:14:25	728	00:14:25	743	00:14:25	608	00:14:25	708	00:14:25	689
26	00:15:00	729	00:15:00	744	00:15:00	609	00:15:00	709	00:15:00	690

Figure 8. Experimental results of MSTSP Accuracy

The system was tested repeatedly for 30 minutes and during the test all nodes, including the heterogeneous nodes, were switched off and on. The motes sent time stamp messages to each other for 30 minutes. The results are shown in figure 8 and demonstrate that our MSTSP works accurately.

We also evaluated our MSTSP by comparing it with Zigbee in terms of sleep time and power saving. The experimental results demonstrate that our MSTSP is more efficient than Zigbee in the environment of heterogeneous USNs.

Figure 9 shows the results of comparing the sleep time of our MSTSP with that of Zigbee. In future research, we will consider the size of each cluster for efficiency and we will perform experiments according to the size of the cluster. Also, the appropriate numbers of slaves under the one master node will be determined for efficient network routing.

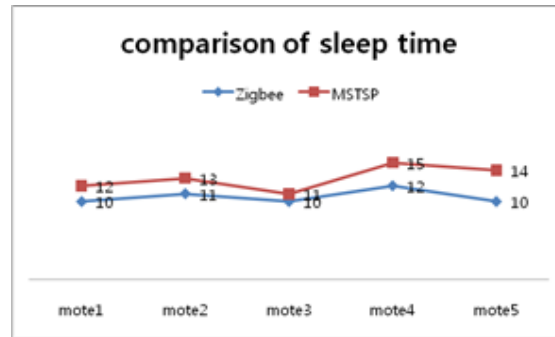


Figure 9. Experimental results of MSTSP Energy Efficiency

## 5. Conclusions

USN has limitations in terms of the source of power and time synchronization of different kinds of sensor nodes. We applied the concepts of master-slave topology to the time synchronization of the heterogeneous USN and proposed MSTSP (Master-Slave Time Synchronization Protocol). We implemented two-phase time synchronization algorithms using both an extended FTSP for the homogenous sensor nodes and MSTSP for the heterogeneous sensor nodes. We then demonstrated that MSTSP works well through simulated USN experiments with several heterogeneous motes.

The contribution of this paper is that the proposed MSTSP can provide not only time synchronization for heterogeneous sensor nodes but also longevity of energy because it maintains 'power off' during the maximum sleep periods.

## Acknowledgements

This work was supported by Konkuk University.

## References

- [1] G. Asada, M. Dong, T. Lin, F. Newberg, G. Pottie, W. Kaiser and H. O. Marcu, "Wireless integrated network sensors: low power systems on a chip", in Proc. Of the 24th European Solid-State Circuits Conference, The Hague, Netherlands, (1998) September.
- [2] M. Lemmon, J. Ganguly and L. Xia, "Model-based clock synchronization in networks with drifting clocks", in Proc. Of the 2000 Pacific Rim International Symposium on Dependable Computing, Los Angeles, CA, (2002) December.
- [3] R. Ostrovsky and B. Patt-Shamir, "Optimal and efficient clock synchronization under drifting clocks", in Proc. of the 18th ACM Symposium on Principles of Distributed Computing (PODC1999), (1999).
- [4] T. K. Srikanth and S. Toueg, "Optimal clock synchronization", J-ACM, vol. 34, no. 3, (1987), pp. 625-645.
- [5] D. L. Mills, "Internet time synchronization: the network time protocol", IEEE Trans. Communications, vol. 39, no. 10, (1991), pp. 1482-1493.
- [6] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks", in Proc. Of the 2001 International Parallel and Distributed Processing Symposium(IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, (2001) April.
- [7] K. Romer, "Time synchronization in ad hoc networks", in Proc. Of ACM Mobihoc, Long Beach, CA, (2001) December.
- [8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", published by Elsevier Science, (2001).
- [9] G. Zheng, L. Zhang and J. Li, "An Energy Supply System for Wireless Sensor Network Nodes", vol. 2012, Article ID 603709, (2012).
- [10] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", in Proc. of the ACM MobiCom'00, Boston, Ma, (2000), pp. 55-67.

- [11] C. Jaikao, C. Srisathapornphat and C. Shen, "Diagnosis of sensor networks", IEEE International conference on Communications ICC'01, Helsinki, Finland, (2001) June.
- [12] H. Yang and B. Sikdar, "A Protocol for tracking mobile targets using sensor network", IEEE Workshop on Sensor Network Protocols and Applications, (2003) May.
- [13] J. Elson, L. Girod and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", in UCLA Technical Report 020008, (2002) February.
- [14] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The Flooding Time Synchronization Protocol", In the proc. Of Sensys'04, Baltimore, Maryland, (2004).
- [15] W. Ye, J. Heidemann and D. Estrin. "An energy efficient mac protocol for wireless sensor networks", In IEEE INFOCOM, (2002).
- [16] K. J. Jeong and M. S. Shin, "The study of time synchronization protocol in USN", FITAT 2011, (2011).

## Authors



**Yongwan Ju**

He received his Ph. D. degree at the Dept. of Computer Engineering from Soongsil University in 2007. From 2002 to 2004, he was an APNIC Executive Council Member. He was an exchange professor in the KIMEP University, Kazakhstan, from 2007 to 2008. He is currently working as a Vice President at the Industry Development Division in the Korea Internet & Security Agency.



**Woojin Paik**

He received the PhD in Information Transfer from Syracuse University, USA in 2000. During 2001-2004, he was a faculty member at Dept. of Computer Science, U. of Massachusetts Boston, USA. He is now a professor at Dept. of Computer Engineering, Konkuk University Glocal Campus, Korea.



**Moonsun Shin**

She received her Ph. D. degree at the Department of Computer Engineering from Chungbuk National University in 2004. She is currently working as a professor in Konkuk University since 2005.  
\*Corresponding Author.

