

Estimation and Compensation of Non-deterministic Delays for Time Synchronization in Wireless Sensor Networks

Shiu Kumar¹, Yeonwoo Lee² and Seong Ro Lee³

^{1,3}*Department of Information Electronics Engineering*

²*Department of Information Communication Engineering
Mokpo National University, South Korea*

¹*kumar.shiu@hotmail.com*, ^{2,3}*{ylee, srlee}@mokpo.ac.kr*

Abstract

New research opportunities are emerging as Wireless Sensor Networks (WSNs) are penetrating into the industrial domain; one such prospect is within time synchronization. For the operation of WSNs, partial or full time synchronization is needed. Time synchronization is very important in a distributed WSN for the purpose of data fusion, time interval calculation, context-free access to radio channel, coordination, event ordering and power management. This paper proposes a novel distributed time synchronization protocol for WSNs, which compensates for the nondeterministic delays by estimating the uncertainties in the synchronization packet delays. The main idea is to minimize the uncertainties in the synchronization packet delays to achieve a synchronization scheme that is accurate, efficient and has a low communication overhead. An average per hop synchronization accuracy of 2.4 μ s has been achieved.

Keywords: *Communication Overhead; Non-deterministic Delays; Packet Delays; Timestamp; Time Synchronization; Wireless Sensor Networks*

1. Introduction

Time synchronization of the nodes has been an open research issue in WSNs [1] having a very long history, which is very important for energy conservation [2]. Over the past few decades, different mechanisms and algorithms have been extensively proposed and used. Many protocol design and application implementation requires precise time, for example, data fusion, data suppression, data scheduling, performing acoustic ranging, *etc.* WSN nodes consist of numerous tiny radio devices equipped with tiny sensors that are typically battery powered. Due to continuous technological advancements, hardware is becoming cheaper and hardware power consumption is decreasing, which both make WSN increasingly more viable for use in real-world applications. Currently, WSNs is widely employed for various applications such as monitoring [3-6], surveillance [7-9], industrial and process automations [10], *etc.*

WSNs have several distinctive characteristics that often prevent the use of existing time synchronization protocols in this field. First, the time synchronization scheme must be implemented in an energy-efficient way as the amount of energy available to the sensors is limited since they are battery powered. Second, wireless communication has limited bandwidth which restricts frequent message exchanges among the sensor nodes. Therefore the synchronization protocol must have a low communication overhead. Third, sensor nodes are very small in size thus posing limitations on

computational power and storage space. Although using GPS provides high synchronization accuracy, its drawback is that it consumes a large amount of power making it unsuitable for WSN application. Network time protocol (NTP) is also infeasible as it does not scale well for WSNs and is intended for traditional computer networks.

In order to facilitate time synchronization, two different approaches exist [1]: First - Distributed clocks can be synchronized to an actual time standard like Universal Coordinated Time (UTC), referred to as physical clock. Second - Distributed clocks can be relatively synchronized to one another, referred to as logical clocks. Synchronization is usually based on some sort of message exchange among sensor nodes. However, these message exchanges that take place during synchronization are subject to numerous nondeterministic delays, which in terms of end-to-end single-hop transmission include the send time, access time, transmission time, propagation time, reception time and receive time.

The problem of time synchronization consists of giving all the nodes of the network a universal time scale to operate. This universal time scale is generally achieved by occasionally synchronizing the clock at each node to a reference time source. The local time seen by each constituent of the system is therefore approximately the same. In order to achieve time synchronization in WSNs, the following four challenges have to be faced; nondeterministic delays, drift, robustness, and convergence speed. However, the focus of this paper is the nondeterministic delays. There are various sources of message delivery delays in WSNs and can be of magnitude larger than the required precision of the time synchronization. Hence, compensating for these delays will result in improved synchronization accuracy. Reference [11] describes the components of message latency that is comprised of 4 distinct components plus the granularity of the node's clocks. First, send time is the time used to assemble and issue the send request to the MAC layer on the transmitter side. Second is the access time, which is the delay incurred waiting to access the transmit channel up to the point when transmission of the packet begins. Third, the transmission time is the time taken by the sender to transmit the message. This time depends on factors such as the length of the message and the radio speed. Fourth, the propagation time which is the time it takes for the message to be transmitted from the sender to the receiver once it has left the sender. Fifth, reception time is the time taken for the receiver to receive the message and is same as the transmission time as shown in Figure 1. Sixth is the receive time, which is the time taken to process the incoming message and to notify the receiver.

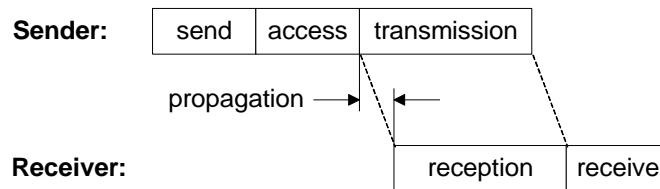


Figure 1. General wireless message delivery decomposition

This paper describes a time synchronization scheme which estimates and compensates for the nondeterministic delays using sensor nodes comprising of Arduino Mega 2560 microcontroller in conjunction with the Nordic NRF24L01+ radio module. The scheme is based on the sender-receiver synchronization employing a single radio message exchange, also allowing for multi-hop synchronization.

The rest of the paper is organized as follows. Section 2 summarizes the related works. Section 3 presents the proposed scheme, which is the non-deterministic delay estimation and compensation scheme. The implementation results and discussions are presented in Section 4. Finally, Section 5 briefly summarizes the results obtained and proposes potential research directions.

2. Related Work

Typical time synchronization algorithms in WSNs include reference broadcast synchronization (RBS) [12], timing-sync protocol for sensor networks (TPSN) [13] and flooding time synchronization protocol (FTSP) [14].

The RBS [12] protocol is based on the receiver-receiver synchronization technique that synchronizes a cluster of wireless sensors within transmission range of the reference sensor node. A third party broadcasts the reference beacon to all the receivers in range. The receivers timestamp the time when the reference beacon is received and hence all the receivers exchange this time information with their neighbouring nodes, which is sufficient information to be able to calculate the offset and synchronize their local clocks. This method eliminates the send time and access times associated with packet delay, however the propagation time and receive time uncertainties still exist. A synchronization accuracy of $11\mu\text{s}$ has been reported for the RBS method implemented on Berkeley motes.

TPSN [13] is a timing synchronization algorithm for sensor networks based on the sender-receiver synchronization comprising of two phases: a level discovery phase and a synchronization phase. The whole network is structured into a hierarchical tree topology with the master node at its core in the level discovery phase. Using the classical sender-receiver handshake exchange, the pairwise synchronization is then performed along the branches of the hierarchical structure in the synchronization phase. To reduce message delivery non-determinism and improve synchronization accuracy, TPSN uses MAC layer time stamping. This method has been developed as a flexible extension of NTP for use in WSNs; however there are few limitations to this approach. The average single hop accuracy of the TPSN method reported is $16.9\mu\text{s}$.

FTSP [14] is a multi-hop sender-receiver synchronization method aimed for resource limited wireless platforms requiring stringent accuracy, using low communication bandwidth. MAC-layer time stamping has been utilized together with error compensation including clock skew estimation. The FTSP method records multiple timestamps on both the sender and receiver, which are made at each byte interval after the SYNC bytes as they are transmitted or received. Several mechanisms are also used to provide robustness against node and link failures. However, it employs a single radio message with only the final error corrected timestamp embedded in it. The method was tested on the Berkeley Mica2 platform and with only 6 timestamps, the average and maximum per-hop synchronization errors were $1.4\mu\text{s}$ and $4.2\mu\text{s}$ respectively.

To mention, a few other time synchronization protocols reported are on-demand synchronization (ODS) [15], clapping and broadcasting synchronization (CBS) [16], delay measurement time synchronization (DMTS) [17], Pairwise lightweight protocol [18], energy efficient and rapid time synchronization (EETS) [19], time diffusion protocol (TDP) [20], global clock synchronization (GCS) [9], average time synchronization (ATS) [21], and consensus clock synchronization (CCS) [22].

3. The Proposed Packet Delay Estimation Scheme

The main goal of the proposed scheme is the estimation and compensation of the nondeterministic synchronization packet delays in order to achieve higher synchronization accuracy. This proposed scheme also employs only a single radio message to synchronize a node's local clock thus also accounting for an energy-efficient and low communication overhead system. The sensor nodes used for this research consists of the Arduino Mega 2560 microcontroller together with the Nordic 16MHz NRF24L01+ single chip 2.4GHz radio module operated at a speed of 250kbps. The radio module comes with an embedded baseband protocol engine, which makes it suitable for low power wireless applications. Having a power down current of 900nA, the module can be configured in various other modes such as receive (RX) mode, transmit (TX) mode, standby-I mode and standby-II mode.

3.1. Proposed system architecture and operation

The overall architecture of the proposed system is shown in Figure 2. Upon initializing, the reference node will request the user to enter the password to gain access to the system. For user friendliness, the keypad and LCD are connected to the system. For user friendliness, the keypad and LCD are connected to the system.

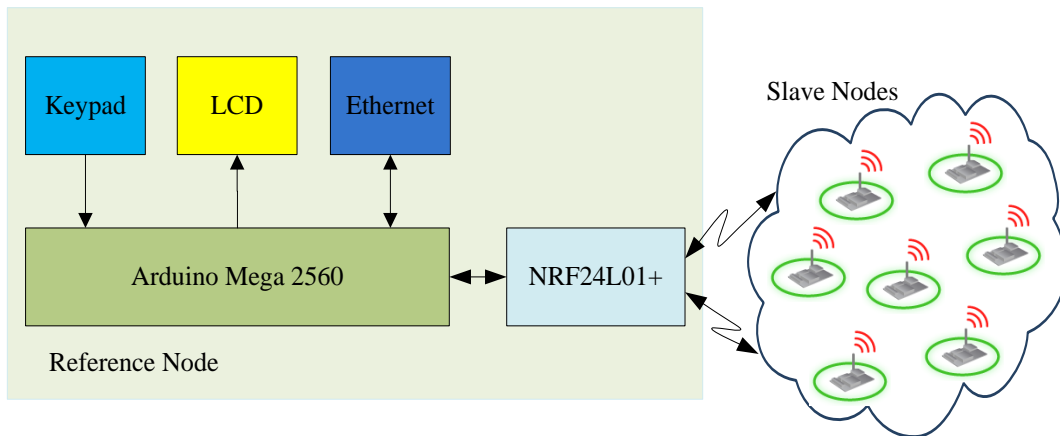


Figure 2. The proposed system architecture

The user enters the password using the keypad and if the correct password is entered, the user will then be given the option whether he/she wishes to enter the time information manually using the keypad or wants to have it done automatically. If manual option is selected, the user will be guided through to enter the time and date information. If the automatic option is selected, the node will retrieve this information over the internet using the Ethernet shield. The user also has the option to change to the desired time zone. Once the time and date information is available to the reference node, it will then synchronize its local clock using the information obtained. After which, the reference node proceeds to send the time stamped information to the slave nodes for the purpose of time synchronization. Other required tasks can also be carried out.

3.2. Synchronization message packet layout

The synchronization message packet layout is one of the important factors that assist in the compensation of the estimated nondeterministic delays in the wireless message delivery. Although the Arduino Mega 2560 microcontroller unit (MCU) has inbuilt timers, a new timer was developed that is capable of keeping track of the date and time information to the accuracy of one clock cycle, that is 62.5ns. Figure 3 shows the general packet layout of the radio message that has been employed where Preamble is a bit sequence used to synchronize the receiver demodulator to the incoming bit stream while Address is the receiver address. The Packet Control Field (PCF) consists of the Payload Length (6 bits), Packet Identification (PID, 2 bits), and the No Acknowledgement Flag (1 bit). The payload contains the timestamp information. Finally, Cyclic Redundancy Check (CRC) which is a mandatory error detection mechanism in the packet and is calculated over the address, Packet Control Field and Payload.

Preamble 1 byte	Address 5 bytes	PCF 9 bits	Payload 20 bytes	CRC 1 bytes
--------------------	--------------------	---------------	---------------------	----------------

Figure 3. The synchronization radio message packet layout

The payload containing the timestamp information comprises of the year, month, day, hour, minute and seconds that are all 2 bytes long. For the year 13 is used to represent 2013. Then we have the millisecond which is 3 bytes long followed by the 5 bytes TCNT1 value, which contains the 16 bit Timer1 clock count information.

3.3. Transmitter packet delay estimation

The proposed scheme intends to minimize the uncertainties in the synchronization packet delay by estimating and compensating for these nondeterministic delays. In order to achieve this, the timing sequence diagram for the NRF24L01+ radio module shown in Figure 4 has been thoroughly investigated and studied.

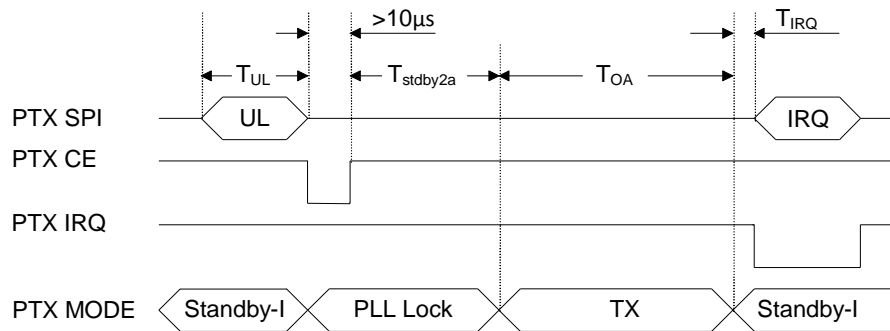


Figure 4. Timing sequence of the NRF24L01+ radio module

The Enhanced ShockBurst mode with its timing controlled via the Data and Control interface has been used together with no acknowledgement. The terms PTX refer to the Primary Transmitter, PTX CE is the Chip Enable pin, PTX mode indicates the mode in which the transmitter is at various stages, PTX IRQ is the interrupt pin, T_{UL} is the time taken to upload the payload to the PTX FIFO and T_{OA} is the time taken to transmit the

radio message over the air. $T_{stdby2a}$ is the time taken for the radio to get into transmit mode while the T_{IRQ} is the time taken for the interrupt to be generated after the data in the PTX FIFO has been transmitted over the air.

Time stamping is performed at the MAC layer and the uncertainties in the PTX are estimated and added to the timestamp just before the packet payload is uploaded to the PTX FIFO. The total transmitter delay, comprising the send, access and transmission time, is given by Equation 1. Where PD_T refers to the transmitter packet delay in number of clock cycles, T_{CE} is the duration for which the PTX CE pin is low and T_t refers to the software processing time of the transmitter.

$$PD_T = \frac{1000}{62.5} \left(T_{UL} + T_{CE} + T_{stdby2a} + T_{OA} + T_{IRQ} + T_t \right) \quad (1)$$

Several experiments were conducted to determine the parameters in Equation 1. Low level programming to output high and low at the MCU output pin as the payload upload started and finished at the PTX FIFO was used and the oscilloscope was used to measure the time taken for different payload size. The result was thus analyzed and the equation to estimate the payload upload time (T_{UL} in μs) for different payload lengths was obtained as given in Equation 2.

$$T_{UL} = 2.06(PL) + 20.48 \quad (2)$$

The value of $T_{stdby2a}$ is given to be $130 \mu s$ [23] while T_{IRQ} for air data rate of 250kbps is given as $17 \mu s$. An oscilloscope has been used to measure the time taken from the time when the PTX CE pin goes high to the time when the interrupt on PTX IRQ pin goes high. The value of the transmission time over the air, T_{OA} (in μs) was thus obtained by subtracting the values of $T_{stdby2a}$ and T_{IRQ} from these measured values and is given by Equation 3.

$$T_{OA} = 32(PL) + 260.15 \quad (3)$$

The time for which the PTX CE pin is low was measured using the oscilloscope and it was found to be a constant value of $21.12 \mu s$. The software processing time of the transmitter, T_t , is the time taken to execute Equation 4, which is the compensation of the nondeterministic delays and is executed after the time stamping. It involves addition of the estimated uncertainties to the timestamp data. The TCNT1 is a 16 bit register, however the clock count for each cycle is restricted to 64000 clock cycles, which corresponds to 4ms.

$$TCNT1 = TCNT1 + PD_T \quad (4)$$

3.4. Receiver packet delay estimation

The receiver delay uncertainties accounts for the propagation time and the receive time which includes the time taken to validate the received message. The reception time is the same as the transmission time and has already been compensated for at the transmitter. The total receiver delay is given by Equation 5. On the primary receiver (PRX), the time difference between the PTX IRQ and the PRX IRQ denoted as T_{IRQ} , gives the propagation time inclusive of the time taken to validate the received message. This value is independent of the message length; however it varies due to interference present from other sources and other tasks that are being run on the nodes at that instance. The average time was found to be $6.185 \mu s$ with the worst case error of $0.575 \mu s$.

$$PD_R = \frac{1000}{62.5} (T_{IRQ} + T_{INT} + T_{recv} + T_r) \quad (5)$$

At the receiver, time stamping is performed as soon as the radio message is validated and available in the PRX FIFO, which is when the PRX IRQ interrupt is generated. An interrupt was generated on the MCU using this interrupt pin to read the time at which the message is received. However, time (T_{INT}) is taken for the MCU interrupt to be generated from the time the PRX IRQ pin interrupt is generated. A hundred samples of this time were recorded using an oscilloscope and an average value of $3.90\mu s$ which is approximately 63 clock cycles was obtained having the worst case error of $0.40\mu s$. Upon performing time stamping, the message is read by the MCU from the PRX FIFO. Another timestamp is taken upon successfully reading the message in order to determine the time taken to receive the message, T_{recv} . T_r is the software processing time of the receiver that is the time taken to execute Equation 6, that is adding the receiver uncertainties to the TCNT1 value of the timestamp received from the transmitter. And finally the slave nodes local clock is synchronized with this modified timestamp value, which compensates for both the transmitter and receiver synchronization packet delays.

$$TCNT1 = TCNT1 + PD_R \quad (6)$$

4. Results and Discussion

The proposed method was implemented on the nodes and experiments were carried out in order to determine the synchronization accuracy of the system. All experiments were carried out in an indoor environment with the presence of interference from Wi-Fi and other radio frequency devices while running multi-tasks to considering the real time environment. The results are also subject to clock drift and skew as compensation algorithm was not implemented. First, only two nodes were used with one being the reference node and the other being the slave node. The receiver node starts off by initializing the radio module and then it waits for the sender's radio message containing timestamp to synchronize its local clock. On the other hand, the sender node also starts off by initializing the radio module. It then starts to send the time stamp message over the radio to the receiver(s) after every 10ms for the purpose of testing the synchronization accuracy. Software algorithm was written, for both reference and slave nodes, to output high at one of the MCU outputs at every timer overflow interrupt. These two output pins were used to measure the synchronization accuracy using an oscilloscope. The experiment was then also carried out for double and triple hops.

Two hundred samples of the synchronization accuracies for single, double and triple hops with all possible payload lengths were recorded and analyzed. The results for the average single hop synchronization accuracy is summarized and compared with other typical time synchronization algorithms as shown in Table 1. The average and the worst case synchronization accuracies are shown in the graph of Figure 5.

Table 1. Comparison of the proposed method with other typical time synchronization protocols in WSNs

	Proposed Method	TPSN	RBS
Average error (in μs)	2.20	16.9	29.13
Worst case error (in μs)	4.64	44	93
Best case error (in μs)	0	0	0
Percentage of time error is less than or equal to average error	56	64	53
Energy Efficiency	High	High	High
Complexity	High	Low	High

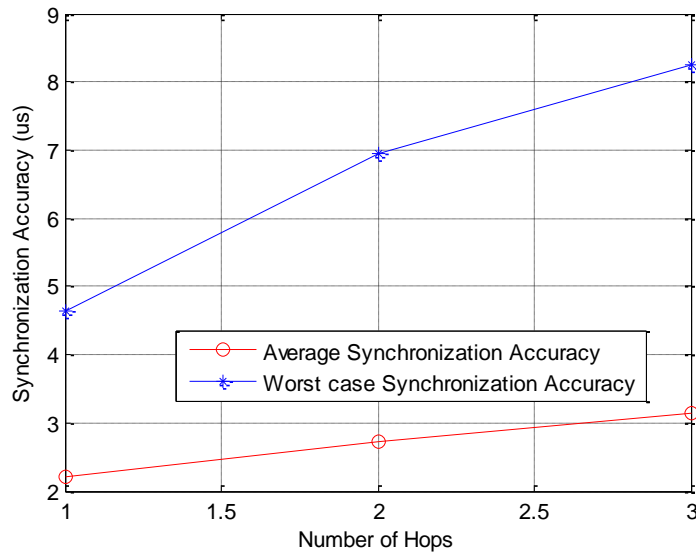


Figure 5. Graph showing the average and worst case synchronization accuracies of the proposed synchronization scheme

5. Conclusion

This research paper proposed a novel clock synchronization scheme, which takes into account the nondeterministic delays. The nondeterministic delays have been estimated and compensated for; achieving a synchronization scheme that employs only a single radio message to synchronize the slave nodes local clock having enhanced synchronization accuracies compared to most representative clock synchronization protocols such as RBS and TPSN.

Furthermore, since only one-way message exchange is required for clock synchronization, the scheme significantly reduces the overall energy consumption of the nodes having a low communication overhead. This is another advantage as wireless sensor nodes need synchronization protocols that are energy-efficient and utilizes a low bandwidth. The proposed scheme is also scalable for multi-hop systems.

However, further work is recommended forming the sensor node using Arduino FIO and NRF24L01+ radio module and evaluating its performance against other existing

sensor nodes that utilizes XBEE radio, also operating at 2.4GHz frequency band. It is also suggested that the clock drift and skew algorithms be implemented with the proposed scheme.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2009-0093828).

References

- [1] B. Sundararaman, U. Buy and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, vol. 3, no. 281, (2005).
- [2] G. Anastasi, M. Conti, M. D. Francesco and A. Passarella, "Energy conservation in wireless sensor networks: a survey. *Ad Hoc Networks*, vol. 7, no. 537, (2009).
- [3] H. -L. Fu, H. -C. Chen and P. Lin, "APS: Distributed air pollution sensing system on Wireless Sensor and Robot Networks", *Computer Communications*, vol. 35, no. 1141, (2012).
- [4] H. -L. Fu, H. -C. Chen and P. Lin, "APS: Distributed air pollution sensing system on Wireless Sensor and Robot Networks", *Computer Communications*, vol. 35, no. 1141, (2012).
- [5] S. Bhattacharjee, P. Roy, S. Ghosh, S. Misra and M. S. Obaidat, "Wireless sensor network-based fire detection, alarming, monitoring and prevention system for Bord-and-Pillar coal mines", *Journal of Systems and Software*, vol. 85, no. 571, (2012).
- [6] B. -R. Son, S. -C. Shin, J. -G. Kim and Y. -S. Her, "Implementation of the Real-Time People Counting System using Wireless Sensor Networks", *International Journal of Multimedia and Ubiquitous Engineering*, vol. 2, no. 18, (2007).
- [7] S. K. Singh, M. P. Singh and D. K. Singh, "Intrusion Detection Based Security Solution for Cluster-Based Wireless Sensor Networks", *International Journal of Advanced Science and Technology*, vol. 30, no. 83, (2011).
- [8] M. S. Islam and S. A. Rahman, "Anomaly Intrusion Detection System in Wireless Sensor Networks: Security Threats and Existing Approaches", *International Journal of Advanced Science and Technology*, vol. 36, no. 1, (2011).
- [9] L. Qun and D. Rus, "Global clock synchronization in sensor networks", *IEEE Transactions on Computers*, vol. 55, no. 214, (2006).
- [10] Y. Guo, "Energy attenuation-based time synchronization for mobile underwater sensor networks", *The Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 57, (2012).
- [11] H. Kopetz and W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems", *IEEE Transactions on Computers*, vol. 36, no. 933, (1987).
- [12] J. Elson, L. Girod and D. Estrin, "In ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation", vol. 36, (2002), pp. 147-163.
- [13] S. Ganeriwal, R. Kumar and M. B. Srivastava, "In Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03)", (2003), pp. 139-149.
- [14] M. Maróti, B. Kusy, G. Simon and Á. Lédeczi, "In Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)", (2004), pp. 39-49.
- [15] Z. Zhong, P. Chen and T. He, "In IEEE INFOCOM", (2011), pp. 2480-2488.
- [16] X. Shen, X. Qian, B. Zhao, Q. Fang and G. Dai, "Clapping and Broadcasting Synchronization in Wireless Sensor Networks", *Tsinghua Science & Technology*, vol. 16, no. 632, (2011).
- [17] S. Ping, "Delay Measurement Time Synchronization for WSNs", *Intel Research*, vol. 59, no. 11, (2003).
- [18] J. V. Greunen and J. Rabaey, "In WSNA '03 Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications", (2003), pp. 11-19.
- [19] B. -K. Kim, "Energy-efficient and rapid time synchronization for wireless sensor networks", *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2258, (2010).
- [20] W. Su and I. F. Akyildiz, "Time-Diffusion Synchronization Protocol for Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, vol. 13, no. 384, (2005).
- [21] L. Schenato and F. Fiorentin, "Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks", *Automatica*, vol. 47, no. 1878, (2011).
- [22] M. K. Maggs, S. G. O'Keefe and D. V. Thiel, "Consensus Clock Synchronization for Wireless Sensor Networks *IEEE Sensors 12*", (2012).
- [23] N. S. ASA, "nRF24L01+ Product Specification: Datasheet", (2008).

Authors



Shiu Kumar

He received Bachelor of Engineering Technology and Postgraduate Diploma in Electrical and Electronics Engineering from the University of the South Pacific in year 2009 and 2012 respectively. Currently he is pursuing Masters degree in Electronics at Mokpo National University, South Korea. His research interests include Automation and Control, Wireless Sensor Networks, Embedded Microprocessor Applications and Digital Electronics.



Yeonwoo Lee

He is currently a Professor with the School of Information Engineering at the Mokpo National University, Mokpo, Korea, since September 2005. He has been a Senior Researcher with 4G Mobile Communication team at the Samsung Advanced Institute of Technologies (SAIT), Kiheung, from January 2004 to August 2005. From October 2000 to December 2003, he has been a Research Fellow with the School of Electronics and Engineering at the University of Edinburgh, UK. From October 2000 to December 2002, he joined core 2 work of Mobile VCE program in UK. He received a MS and Ph.D. in Department of Electronics Engineering from Korea University, Seoul Korea, in 1994 and 2000, respectively. His research interests are wireless multimedia mobile telecommunication systems, radio resource management, (ad-hoc) multi-hop relay system, sensor network and particularly their applicable issues to 4G mobile communication systems and cognitive radio systems.



Seong Ro Lee

He received th B.S. degree in electronics engineering from Korea University, Seoul, Korea, in 1987, respectively, and the M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, 1990 and 1996, respectively. In September 1997, he joined the Division of Electronics Engineering, Mokpo National University, Jeonnam, Korea. His research interests include digital communication system, mobile and satellite communications system, applications of telematics, USN and embedded system. He serves as chairman of detection and estimation committee for the Korea Information and Communications Society.