International Journal of u- and e- Service, Science and Technology
Vol.8, No.1 (2015), pp.267-276
http://dx.doi.org/10.14257/ijunesst.2015.8.1.24

Energy-balanced Time Synchronization Algorithm Based on Flooding for Multi-hop Wireless Sensor Networks

Yang Wang

School of Electronics and Communication Engineering, Shenzhen Polytechnic
Shenzhen, 518055, China
wyang@szpt.edu.cn

Abstract

To achieve network-wide time synchronization in multi-hop wireless sensor networks (WSN), this paper proposes an accurate and energy-balanced time synchronization algorithm based on flooding. Flooding Time-Synchronization Protocol (FTSP) has advanced features, such as implicitly dynamic topology and high time accuracy, which can't be affected by some node's losing. FTSP propagates its time information of the reference node without concern about the network topology. However, in large-scale WSN applications, the efficiency of FTSP will be reduced drastically because of serious network storm. Meanwhile, the energy consumption of the network breaks the energy balance. In this paper, our aim is to solve the problems of synchronization error accumulation and unbalanced energy consumption. Our theoretical findings and experimental results show that forbidding possibly invalid forwarding among the sensor nodes drastically improves the synchronization accuracy and performance of energy-balanced.

Keywords: *wireless sensor network, time synchronization, energy-balanced, Flooding Time Synchronization Protocol, multi-hop*

1. Introduction

Wireless sensor networks (WSNs) can be applied to a wide range of potential applications in domains: area monitoring, health care monitoring, air pollution monitoring, intelligent traffic, natural disaster prevention, environmental monitoring [1-3], and so on. When designing WSNs, there are many important factors to be considered such as tolerance to node failure, dynamitic network topology, hardware constraints, power consumption, and time synchronization [4-6]. Time synchronization is a procedure for providing a common notion of time across a distributed system [7-10]. It is crucial for WSNs when performing a number of fundamental operations, such as:

- **Data fusion.** Data merging is a major operation in all distributed networks for processing and integrating the collected data in a meaningful way, and it requires some or all nodes in the network to share a common timescale [11].
- **Power management.** Energy efficiency is a key factor when designing WSNs since sensors are usually left unattended without maintenance and battery replacement for lifetimes after deployment. Most energy-saving operations depend on time synchronization [12-13]. For instance, sleep and wake-up modes control help the nodes to save huge energy resources by minimal power cost during the sleep mode.
- **Transmission Scheduling.** Time synchronization is required by many scheduling protocol in WSNs [14-15]. As traffic load increases, network throughput drastically

ISSN: 2005-4246 IJUNESST

degrades because the probability of collisions rises, the time division multiple access (TDMA), one of most popular communications schemes for distributed networks only can work based on the synchronized network.

Each sensor node is equipped with a local clock, $LC_i(t)$ which is initialized to $\theta_i(t_0)$. The local clock progresses in accordance with its hardware clock tick ω_i, and k is constant value depends on the physical properties of crystals. Even if all nodes start with the same clock value, the local clocks of the nodes will drift apart from each other as time passes, called clock drifts. The clock model is defined as

$$LC_i(t) = k\int_{t_0}^{t} \omega_i(t)dt + \theta_i(t_0) \tag{1}$$

There are two main factors which account for the synchronization deviation in WSNs' nodes: the frequency instability and inaccuracy of crystal oscillator. Because different manufacturers have differences in the manufacturing process of the crystal oscillator, and the oscillation frequency of the crystal oscillator are vulnerable to a variety of factors affected, such as the voltage, temperature and crystal aging and so on, there is a certain difference between the real oscillation frequency from its nominal frequency of the crystal oscillator, which leads to the node timing error and defined as

$$-\rho_{\max} \le \rho(t) \le \rho_{\max} \tag{2}$$

Uncertainty of sending and receiving delay of wireless link are as followed (the flow chart in Figure 1):

- Sending preparing time. Time of sending node constructing one single message, including kernel protocol processing and buffer time, depends on the consumption of system call and the current load of processor.
- Channel access time. Time of waiting for the transmission channel is idle—the delay from the waiting channel idle to the transmission start, which depends on the current load of network.
- Transmitting completion time. Time used for sending nodes transmitting by bit, which depends on the length of message and the speed of transmitting.
- Propagation time. Time used for spreading between two nodes through media, which primary depends on the distance between nodes.
- Receiving time. Time of receiving node receiving messages by bit and transmit to MAC layer.
- Application preparing time. Time used for the receiving node unpacking the message and transmitting to the application layer.

Figure 1. Sending and Receiving Delay Flow Chart

In the above context, the essential key problems of time synchronization should be considered and satisfied are as followed:

- Improvement of synchronization energy efficiency. The main character of WSNs is the problem of energy limitation of nodes, so the time synchronization method should consider the efficient energy of sensor nodes firstly, and reduce the time needed by synchronizing.
- Improvement of Expansibility. Time synchronization mechanism should support the effective expansion of the number of nodes in the network or density.
- Precision. Many projects require high precision, including the timestamp of video and voice data, and the timestamp of alarm information, especially used in updating the synchronization of deviation of domestic clock.
- Robustness. The WSNs application may be located in unmaintained area for a long time, once some nodes lose effectiveness, the time synchronization mechanism are expected to keep the efficiency and robustness.
- Synchronization limit (or lifespan) Option. The time synchronization method not only supports temporary rule, but permanent one, which is as long as the lifespan of network.
- Rationality of effective synchronization range. Time synchronization regional range area can be geography range of physics or logic range, for example the number of hops of network routing. Time synchronization method can set the range of effective area.

Since the WSNs time synchronization research proposed in 2001, there have been a variety of time synchronization algorithms and protocols, for example DTMS, RBS, TPSN, GPS and other [16-23]. They are different from each other at the performance and mechanism of synchronization. The FTSP this paper mentioned (Flooding Time Synchronization Protocol) is proposed by Branislav Kusy of Vanderbilt University firstly [24]. FTSP method realizes time synchronization between sender and receiver by single broadcast message [25-28]. In considering several requirements in practical application, this paper improved the method, and realizes the time synchronization requirements of multi-hop network nodes.

2. Flooding Analysis

The Flooding protocol belongs to sender-receiver protocol, so sender which is based on the technology of MAC layer timestamp embedded in the sending moment, while receiver also records the receiving moment in the MAC layer. Differing from other sender-receiver synchronization protocol, in the FTSP protocol, sender marks several timestamps sequentially when sending a message. According to these timestamps, receiver could estimate the time delayed by the interrupt. The receiver could compensate for the receiving time according to the time of the interrupt delay and coding and decoding time, and then more precise synchronization points will be obtained (the time stamp in Figure 2), as follows:

- After completion of sending preamble bytes, the FTSP algorithm marks the timestamp (Ts) in time synchronization message and transmits. The timestamp Ts is the current time minus the transmitting time of data part which including timestamp Ts, which can be obtained by the length of data and the speed of transmitting. Ts1 and Ts2 mean the multiple time stamps in sender.
- The receiver records the last bit of preamble bytes reaching time (Tr), and calculates the bit offset. After the complete message received, the receiver calculates the time

delay Td through bit offset, which can be calculated by bit offset and the speed of receiving. Tr1 and Tr2 mean the multiple time stamps in receiver.

- The receiver calculates the time deviation $T_{offset}(t)$ from the sender:

$$T_{offset}(t) = T_r(t) - T_d(t) - T_s(t) \quad (3)$$

and then calculates and adjusts time synchronization between local time and sender's, according to linear regression.

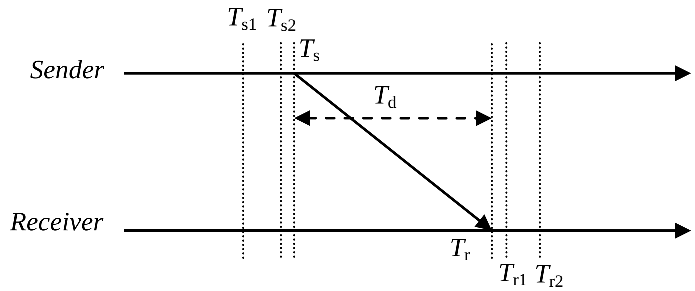

Figure 2. Sender and Receiver Message Time Stamp

To define WSNs time synchronization period as Psp, this Psp is constant value.

$$LC_{ref}^{m+1} - LC_{ref}^{m} = P_{sp} \quad (4)$$

The m represents the No. m times time synchronization sending, LC_{ref}^{m} represents the local time of time reference node when sending the No. m times time synchronization message. η_c is used as measuring deviation degree between clock and perfect clock, defines as (5):

$$\forall t_2 > t_1 \geq 0,\ \eta_c = \left| \beta \times \left| \frac{LC_i(t_1) - LC_i(t_2)}{t_2 - t_1} \right| - 1 \right| \quad (5)$$

β is the relationship function coefficient to change the clock $LC_i(t)$ to real time. The time synchronization in WSNs is defined as (6),

$$\forall t, \left| LC_i(t) - LC_{ref}(t) \right| \leq \delta \quad (6)$$

which means δ -synchronization. $LC_i(t)$ means the local clock of node i in WSNs, and $LC_{ref}(t)$ means the local clock of reference node. δ -synchronization is referred as "synchronization". If the two clocks of a group of clocks are δ -synchronization, then these groups' clocks would be considered as "already synchronized". The concept of δ -synchronization does not need perfect clock, whose clock time don't keep pace with the real time. Because of the wireless transmission rate, the time of time synchronization data transmission τ_d can be considered as a constant. In this assumption, assuming σ representative of τ_w boundary, so the inequality $\tau_w \leq \sigma < P_{sp} - \tau_d$ is established. Thus, the deviation ξ of the reference node and other nodes clock satisfies inequality (7),

$$\begin{cases} \xi \geq \tau_d + \eta_c \left(P_{sp} + \tau_d \right) \\ \xi < \tau_d + \sigma + \eta_c \left(P_{sp} + \tau_d + \sigma \right) \end{cases} \quad (7)$$

As in (7), the minimum value of clock error is as follows:

$$\xi_{min} = \tau_d + \eta_c \left(P_{sp} + \tau_d \right) \quad (8)$$

If the node transmit the time synchronization message without instruction execution time and uncertain access time, $T_d = \tau_d$, but the fact is $T_d > \tau_d$. So through calculating the bit offset, to get real Td.

To define work states of WSN sender as follows:

- **IDLE state:** The micro controller unit (MCU) of WSN node is free and has no air data to deal with.
- **APP-send state:** The node enters the APP-send state when the application layer of WSN node completes packaging the message, and will not quit this state until the whole message transferred into MAC layer.
- **MAC-send state:** When the message transferred into the MAC layer, the node enters into MAC-send state, and will not quit until the whole message transferred into the physical layer.
- **PHY-send state:** Node enters the PHY-send state after the whole message received from MAC layer.

And receiver of WSN also contains four states as follows:

- **IDLE state:** The micro controller unit (MCU) of WSN node is free and has no air data to deal with.
- **PHY-receive state:** From the first byte received beginning from the air to the last bit, node enters into PHY-receive state until all data transmitted into MAC layer after decoding, and then quits the PHY-receive state.
- **MAC-receive state:** After receiving all data packets from physical later, node enters into MAC-receive state, and it will not quit the MAC-receive state until all the data have been transmitted to app layer.
- **APP-receive state:** The nodes will enter into the APP-receive state after all the data from the MAC layer received, and start to unpack the frame data. Then nodes begin to realize the function according to the message until the MCU finishes all jobs and enters into IDLE state.

The main feature of the FTSP protocol is stability, which ensures the whole network healthful when some nodes lose effectiveness. The FTSP adopts the method of flooding to broadcast the time of time reference node, instead of founding topology, and it will broadcast a message periodically if the time reference node works normally in the network. The message include sequence number, each once the time reference node sends a new message, the value of sequence number plus 1, so it can be understood as synchronization rounds. According the value of sequence number in the message, the node can judge the validity of the message. And if the message is a valid new one, it will record the new synchronization time at the buffer, otherwise abandoned if it is invalid. In the network, the node which has received a time synchronization message will calculate the current time of time reference node, and organizes and broadcasts a new time synchronization message to synchronize the other nodes. This process is repeated iteratively and finally all the nodes in the network have synchronized with the time reference node.

While the time reference node fails, the other nodes will not receive any time synchronization messages for a period of time. According to the FTSP protocol, some nodes will be promoted to the time reference node automatically. Through competition, the node with the smallest ID number will become a new time reference node, which protects the stability of the protocol.

The following is the FTSP realization based on one synchronous timestamp.

(1) interrupt service routine(void) {//realizing the MAC layer timestamp in interrupt service and the time offset of byte alignment}

(2) if MCU state == **PHY-receive state**, MCU enters into data packet receiving state;

(3) if MCU state == **APP-send state**, MCU enters into sending state;

- if the preamble byte is not finished yet, MCU continues to send preamble byte;
- if the first time synchronization byte still not be sent yet, MCU continues to send the synchronization byte;
- if the second time synchronization byte still not be sent yet, MCU records the timestamp T_s, and then sends the synchronization byte;

(4) if MCU state == **IDLE state**, MCU enters into idle state;

- if MCU continuously receiving preamble byte, MCU will be ready to receive synchronization byte;
- if new data is preamble data, discards;

  else

  if the second time synchronization byte received, MCU state enters into **PHY-receive state,** and get local time as timestamp T_r;

  if MCU received all time synchronization bytes, then calculates bit offset delay, and repair local clcok.

3. Improved Flooding Design

In practical multi-hop network application, FTSP algorithm will mark the timestamp at the key positions of each message and broadcast to the whole network after selecting a root node. Each nodes of the network will continue to broadcast to the other nodes of the network when they have received the timestamp. Although this method of broadcasting can synchronize the network, the energy consumption and efficiency are very low. Especially in large-scale wireless sensor networks, or under the high-density distribution scenarios, the network storm would bring down the network. To solve this problem, the current FTSP algorithm should be improved to make sure no network storm anymore, and reduce the energy consumption as well. Since the most significant feature of the FTSP is regardless of topology of the network, the FTSP can't be repaired depending on the network topology.

This paper proposed one improved FTSP method as follows:

- The root node broadcasts time synchronization message to the entire network with sequence number and its own location.
- Every node, which can receive the synchronization message of root node in whole network, repairs own clock directly according to the FTSP algorithm.
- After repaired, each node forward synchronization message with sequence number and its own location according to the air conflict rule.

- At the same time, each node keep monitoring and marking messages continuously, marking all FTSP message received and the node address of every frame data.
- When the entire network quiet, every node organizes its own node candidate set, which represent the wireless network scope of the node, and is used to determine whether necessary to forward the time synchronization message.
- The candidate set will be repaired by every time synchronization process after the first candidate set construction.
- After the starting of new time synchronization message, each node executes the FTSP according to the step 2. During the process of waiting the air idle slot, the nodes keep monitoring the entire network, and then judge the received time synchronization message according to the FTSP algorithm and delete the node number contained in time synchronization message from node candidate set.
- The node should determine if any node left in its own candidate set before sending when obtaining the air idle slot. It should continue to send the FTSP time synchronization message if any node left in candidate set, otherwise stop forwarding time synchronization if there is no candidate node at all.

This paper designed four application experiments as follows.

1) 30 wireless sensor nodes uniformly distributed in 100 square meters.
2) 50 wireless sensor nodes uniformly distributed in 200 square meters.
3) 100 wireless sensor nodes uniformly distributed in 300 square meters.
4) 200 wireless sensor nodes uniformly distributed in 400 square meters.

The transmit distance of the wireless sensor nodes adopted in the contrast test is about 30 meters. And this paper measured the real situation by traditional FTSP and improved FTSP in these four application scenarios. In these measurements, this paper designs the synchronization period of 10 seconds, and 50 times synchronization. The results are as follows.

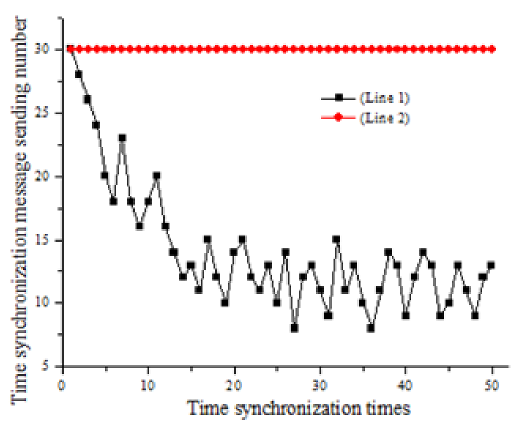

Figure 3. 30 Nodes FTSP Application Result

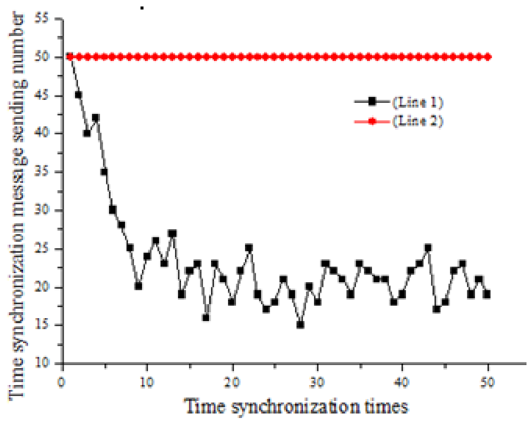

Figure 4. 50 Nodes FTSP Application Result

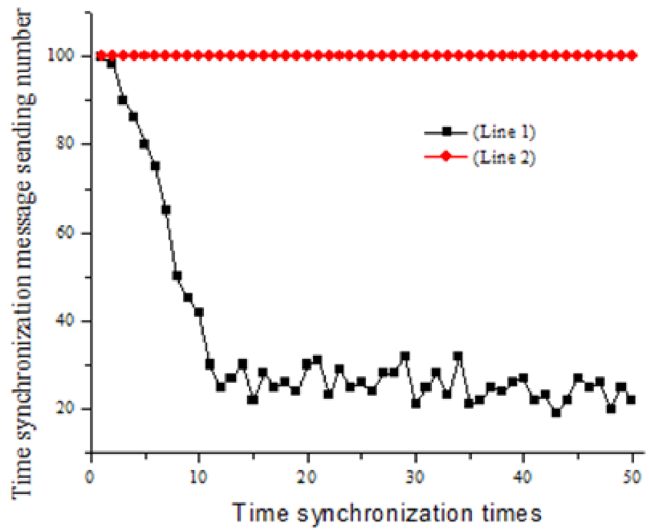

Figure 5. 100 Nodes FTSP Application Result

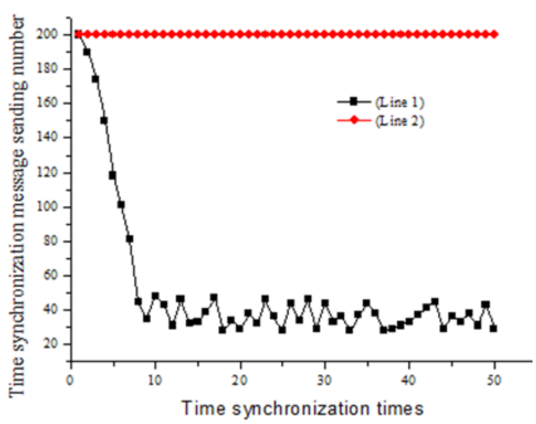

Figure 6. 200 Nodes FTSP Application Result

From Figure 3 to Figure 6, the "Line 1" represents the result of improved FTSP, and the "Line 2" represents the result of original FTSP. Apparently, after FTSP improved, the number of FTSP time synchronization message sending was reduced greatly.

4. Conclusion

In multi-hop WSNs, the time synchronization algorithms should pay more attention to the efficiency of time synchronization message sending. In some multi-hop applications, the flood mode of FTSP may lead the WSNs into network storm, which will reduce the efficiency of network communication greatly. This paper improved the FTSP which can avoid the network storm and reduce the energy consuming. There are four multi-hop application experiments designed in this paper. In these contrast experiments, we can see that, the improved FTSP can reduce the number of time synchronization message sending greatly than the original FTSP, especially in high-density WSN applications, which can reduce the probabilities of network storm efficiently. With the number of message sending reduced, the energy consuming reduces at the same time, which is helpful to prolong the lifetime of network.

Acknowledgements

The paper is supported by the Natural Science Foundation of Guangdong Province of China (No. S2013010014188) and the Basic Research Foundation of Shenzhen of China (No. JCYJ20120618094355291).

References

[1] J. Li, S. Liu and S. Wu, "A Design of Remote Computer House Monitoring and Control System Based on ZigBee WSN", IJACT, vol. 4, no. 12, (**2012**), pp. 233-240.
[2] Z. A. A. Al-Marhabi, L. Fa, F. Zeng, M. Y. A. Alfathi and A. Radman, "Achieving WSN Performance and Forest Monitoring System With WSC", IJACT, vol. 4, no. 12, (**2012**), pp. 77-84.
[3] F. Hui, X.-m. Zhao, X. Shi and J.-y. Zhang, "A Multi-interface WSNs Based Hazardous Materials Transportation Monitoring System", JDCTA, vol. 6, no. 5, (**2012**), pp. 255-263.
[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey, Computer Networks", The International Journal of Computer and Telecommunications Networking, vol. 38, no. 4, (**2002**), pp. 393-422.
[5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", SIGOPS Oper. Syst. Rev., vol. 36, (**2002**), pp. 147–163.
[6] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks", Proceedings of the 1st International Conf. on Embedded Networked Sensor Systems. ACM Press, (**2003),** pp. 138-149.
[7] M. Jadliwala, Q. Duan, S. Upadhyaya, and J. Xu, "Towards a theory for securing time synchronization in wireless sensor networks", WiSec '09. ACM, (**2009**), pp. 201–212.
[8] P. Ferrari, A. Flammini, S. Rinaldi, and G. Prytz, "Time synchronization concerns in substation automation system", Proc. IEEE Int. Workshop AMPS, (**2011**), pp. 112–117.
[9] D. Fontanelli and D. Macii, "Accurate time synchronization in PTP-based industrial networks with long linear paths", Proc. ISPCS, (**2010**), pp. 97–102.
[10] B. Sundararaman, Ugo Buy, Ajay D Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey", Ad Hoc Networks, vol. 3, no. 3, (**2005),** pp. 281-323.
[11] M. Liu, G.-x. Liu and X.-l. Xu, "Distributed Data Fusion Algorithm with Low Energy Consumption and High Accuracy in WSN", IJACT, vol. 3, no. 11, (**2011**), pp. 122-129.
[12] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks", Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing, San Francisco, California, USA, (**2001**) April.
[13] N. Amit, S. Rajesh, N. Meenakshi, L. K. Awasthi, "Wait and Watch Technique: Removing Time Synchronization overhead in Wireless Sensor network", International conference on Advance computing (ICAC) , Thapar University, India, (**2009**) March.

[14] V. Classo, H. Lonn and N. Suri, "Efficient TDMA Synchronization for Distributed Embedded Systems", 20th symposium on Reliable distributed Systems (SRDS), (**2001**) pp. 198-201.
[15] T. Salonidis and L. Tassiulas, "Asynchronous TDMA ad hoc networks: scheduling and performance," Euro.Trans.Telecomms., vol. 15, (**2005**), pp. 391-403.
[16] X. Zhang, X. Tang and J. Chen, "Time synchronization of hierarchical real-time networked CNC system based on Ethernet/Internet", The International Journal of Advanced Manufacturing Technology, vol. 36, no. 11, (**2008**), pp. 1145-1156.
[17] E. A. Lee and S. Matic, "On determinism in event-triggered distributed systems with time synchronization", Proceedings of Precision Clock Synchronization for Measurement, Control and Communication, Washington, D.C., USA: IEEE, (**2007**), pp. 56-63.
[18] K. Arvind, "Probabilistic clock synchronization in distributed systems", IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 5, (**1994**), pp. 474-487.
[19] J. Elson, L. Girod and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts", Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, Boston, MA, USA: ACM Press, (**2002**), pp. 147-163.
[20] M. Akar and R. Shorten, "Distributed probabilistic synchronization algorithm for communication networks", IEEE Transactions on Automatic Control, vol. 53, no. 1, (**2008**), pp. 389-394.
[21] H. R. Carlos and K. Thomas, "A mutual network synchronization method for wireless ad hoc and sensor networks", IEEE Transactions on Mobile Computing, vol. 7, no. 5, (**2008**), pp. 633-646.
[22] S. S. Thaker, S. Nandi and D. Goswami, "Clock synchronization in IEEE 802.11 ad hoc networks", Lecture Note in Computer Science, vol. 3356, no. 1, (**2004**), pp. 180-189.
[23] A. S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks", IEEE Transactions on Information Theory, vol. 52, no. 6, (**2006**) pp. 2725-2748.
[24] J. Xiong, L. Xuan, T. Huang and J. Zhao, "Novel Covert Data Channel in Wireless Sensor Networks Using Compressive Sensing", Journal of Networks, vol. 7, no. 10, (**2012**), pp. 1523-1529.
[25] H. Huang, H. Chen, R. Wang, Q. Mao and R. Cheng, "Secret Sharing Scheme Based on Cylinder Model in Wireless Sensor Networks", Journal of Networks, vol. 7, no. 7, (**2012**), pp. 1009-1016.
[26] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The flooding time synchronization protocol", Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, (**2004**).
[27] T. Kunz, "Implementing clock synchronization in WSN: CS-MNS vs. FTSP", 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), (**2011**), pp. 157- 164.
[28] A. Nayyer, M. Nayyer and L. Kr. Awasthi, "A Comparative study of Time Synchronization Protocols in Wireless Sensor Network", International Journal of Computer Applications, vol. 36, no.11, (**2011**) December, pp. 13- 19.

Author

Yang Wang, He received his B.Eng. in Electronic engineering (1998) and PhD in Mechanical manufacturing and automation (2003) from Dalian University of Technology. Now he is associate professor of School of Electronics and Communication Engineering, Shenzhen Polytechnic. His current research interests include different aspects of Wireless Sensor Networks.