Research Article

Adaptive Compensation for Time-Slotted Synchronization in Wireless Sensor Network

Tengfei Chang¹ and Qin Wang¹,²

¹ University of Science and Technology, Xueyuan Road No.30, Haidian District, Beijing 100083, China
² BSAC, University of California, Berkeley, CA, USA

Correspondence should be addressed to Qin Wang; wangqin@ies.ustb.edu.cn

Received 6 October 2013; Revised 11 March 2014; Accepted 11 March 2014; Published 7 April 2014

Academic Editor: Shancang Li

Copyright © 2014 T. Chang and Q. Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since time division multiple access (TDMA) is employed to reduce power consumption, time synchronization is critically important for wireless sensor networks. Packet-based clock correction is one of main methods to maintain synchronization among nodes. However, in many real cases, such as industrial equipment monitoring and diagnosis, there is no need to exchange messages during relatively long time, while highly accurate time synchronization has to be maintained, which results in that additional packets have to be exchanged specifically for synchronization purpose. For addressing this problem, an adaptive compensation method is proposed in this paper to achieve a high synchronization precision without exchanging message frequently. We proved that the adaptive compensation method can achieve a high synchronization precision with much fewer packet exchanges. In addition, the performance of the proposed scheme has been evaluated in three different experimental settings. In the indoor experiment, compared with the synchronization method without using compensation, 98% of the synchronizing packets can be eliminated while maintaining synchronization precision at 100 μs. In the outdoor experiment with higher temperature fluctuation, 80% of the packets can be eliminated with 100 μs synchronization precision. In the case of sudden temperature changes of more than 10°C, the proposed compensation packet frequency regresses to the frequency of the synchronization method without compensation.

1. Introduction

As most wireless sensor networks (WSNs) [1] work with battery power, power consumption becomes a critical constraint for networks’ nodes. Since radio transmitting is major consumer of a node’s energy [2], it is important to minimize the radio-up time when designing WSNs. A good design for a low power WSN should keep the radio off whenever possible and only turn on radio when actively transmitting or receiving packets.

To achieve this goal, synchronization is required. Time synchronization makes two nodes share the same sense of time; thus they can coordinate their radio activity to achieve low power. This is particularly important for battery equipped devices used for industrial application, which usually require more than 5 years of operation life. As one of such synchronization techniques, time slotted channel hopping (TSCH) is adopted by major industrial low power wireless standards such as WirelessHART [3], ISA100 [4], and IEEE802.15.4e [5].

TSCH protocols use an agreed upon transmission schedule between two nodes. Figure 1 shows the case of TSCH in IEEE802.15.4e. Time is split into equal length time slots; the slot frame is defined as a constant number of slots repeated over time. Each slot has an absoluteslotnumber (ASN), which indicates how many slots have elapsed since the network was initialized. Once a node is synchronized, the communication by exchanging packets between pairs of nodes is scheduled for a specific time slot. All packets are transmitted exactly TxOffset after the beginning of a slot. To ensure that the packet is not missed when slight desynchronization happens, Guard Time is used to turn on the radio of the receiving node before the expected arrival of a packet.

Packet-based clock correction is one of main methods to maintain synchronization among nodes. Since packets are transmitted TxOffset after the beginning of a slot, every packet is implicitly timestamped at TX0ffset after the beginning of a slot. A node synchronizes to its time parent
Reference broadcast synchronization (RBS) [7] is a protocol in which synchronization happened between receivers. The sender broadcasts a packet to a set of receivers. The receivers record the timestamp when the packet was received and exchange the timestamps with other receivers in the synchronization set. RBS minimizes the undetermined delay from the sender side which improves the accuracy of synchronization. Time-sync protocol for sensor network (TPSN) [8] is a different protocol with RBS, which records the timestamp in the MAC layer at the sender side. Two communicating nodes exchange packets using a two-way method. This pairwise synchronization can remove the propagation delay. The timestamp is added as a final step after encoding/decoding the SYNC byte so the node can achieve a high accuracy of time synchronization. TSCH protocol in IEEE802.15.4e just uses FTSP as its synchronization method.

Those synchronization protocols focus on high accuracy of synchronization when exchanging packets. To maintain such high accuracy they have to resynchronize frequently to deal with clock drift, which consumes a lot of energy and bandwidth. Some adaptive synchronization algorithms, through estimating drift, adjust the local clock to counteract the clock offset without exchanging packets for resynchronization. By modeling the clock drift of daytime, [10] presents an adaptive synchronization to deal with drift. However, the model is built based on offline measurement. It cannot handle the dynamical variation of drift. Self-correction clock, which is shown in [11], is another method that made a hardware clock which can do self-correction to deal with drift. This obviously increases the cost of node.

The Adaptive Compensation mechanism also focuses on how to maintain high accuracy without exchanging packets but using an advanced simple model. The simplest model for drift is averaging the clock offset in a period, which assumes that the drift is constant. This hypothesis is only true in several minutes or indoor environment. Furthermore, the accuracy of calculated drift is not enough to satisfy industry applications such as industrial equipment monitoring and diagnosis, which usually need to exchange packet infrequently. Additional packets are still required to be exchanged to maintain synchronization. However, averaging the clock offset in a period is simple enough to be used practically. Adaptive Compensation mechanism performs an iteration procedure based on this simple model to gradually increase accuracy of calculated drift. Using this accuracy-increased drift to adjust (compensate) on the local clock, a one-hour synchronization interval can be achieved for indoor case. A temperature sensor-based detecting mechanism is also proposed in Adaptive Compensation to deal with the influence of temperature on clock drift.

3. Problem Definition

In general, compensation can be explained as periodically adjusting the local clock at a given rate, usually an estimated value of node’s drift. For example, a node in a slotted network,
which has a 32768 Hz crystal with a drift of 10 ppm faster than its time parent’s crystal, can keep synchronization by shortening its current time slot duration by one tick (30 µs) every 3 seconds (Figure 2).

Since the temperature is the main factor influencing clock drift [12], drift can be considered constant in short-term or indoor environments where temperature fluctuations are small. In such environments, drift can be calculated by averaging a period of synchronization error as (1) shown. offset indicates the synchronization error, which is achieved by comparing the timestamp withTxOffset. Δt indicates the period since the last packet-based synchronization, called the synchronizing interval:

\[
\text{drift} = \frac{\text{offset}}{\Delta t}. \tag{1}
\]

The accuracy of offset is determined by the accuracy of the timestamp. A 32768 Hz crystal has a timestamp accuracy of 30 µs. The calculated drift cannot counteract the real drift completely with such timestamp accuracy. Packet-based resynchronization is still required. The only effect is lengthening of the interval compared to synchronization without compensation. When Guard Time is given, the length of the resynchronization interval is determined by the accuracy of the calculated drift. One way to achieve higher accuracy is using a high frequency crystal. However a high frequency crystal itself would consume significant energy. Therefore, this solution is not suitable for the application in low power WSNs.

The offset achieved by comparing timestamp withTxOffset can be presented by real offset plus an error, where the error is smaller than one tick (2). Tick is a measurement of time which is determined by the frequency of the crystal. One tick for a 32768 Hz crystal equals 30 µs or 1/32768 of a second. With (2), we substitute the offset in (1) and derive (3):

\[
\text{offset} = \text{offset}_{\text{real}} + \text{error}, \quad |\text{error}| \leq 1 \text{ tick}, \tag{2}
\]

\[
\text{drift} = \frac{\text{offset}_{\text{real}}}{\Delta t} + \frac{\text{error}}{\Delta t}, \quad |\text{error}| \leq 1 \text{ tick}. \tag{3}
\]

As our goal is to improve the accuracy of drift, what we need to do is diminish the value of error/Δt in (3). Enlarging Δt is an obvious solution. Enlarging Δt can be explained as scheduling the next synchronizing packet to have a longer interval. However, excessively enlarging would make the synchronization error fall out of Guard Time before the next synchronization. The next question is how long the interval should be to avoid this falling out. In the following section, we will answer this question based on a model of synchronization error using compensation.

4. Adaptive Compensation

In this section, first we will show Adaptive Compensation in theory based on a model of synchronization error with compensation. Then we propose our Adaptive Compensation synchronization method.

4.1. Model of Synchronization Error with Compensation. The model of synchronization error with compensation is depicted below (4). Dr and Dc indicate the real drift and calculated drift. T indicates time. E_{sync} indicates synchronization error at T. This error can be larger or smaller than zero, which means that the clock of node is in front of or behind its time parent. This model could be explained as that the inaccuracy of drift, which responds to (D_r - D_c), will make synchronization error increase to E_{sync} at T:

\[
E_{\text{sync}} = (D_r - D_c) \times T, \quad \tag{4}
\]

\[
T = \frac{\text{Guard Time}}{D_r - D_c}, \quad \tag{5}
\]

\[
D_r - D_c \leq \frac{1}{\Delta t} \text{ tick}, \quad \tag{6}
\]

\[
T \geq \frac{\text{Guard Time} \times \Delta t}{1 \text{ tick}} = T_{\text{schedule}}. \quad \tag{7}
\]

The key to answer the question is to know what time synchronization error will fall out of Guard Time. With this thought, we transform the model by exchanging the sides of E_{sync} and T and replace E_{sync} with Guard Time. As (5) shows, it is clear to see that T is the time when synchronization error will fall out of Guard Time.

As we have got the time, what we should do is schedule next synchronizing packet before T. However, we did not know the specific time point to guide node to schedule next synchronizing packet since D_r - D_c is unknown.

Actually D_r - D_c is the error/Δt in (3), which is caused by the inaccuracy of drift. Since error is smaller than one tick,
Initial Interval parameters are required: this Adaptive Compensation method in practice, two parameters are required: Initial Interval for synchronizing and Required Accuracy, or the length of Guard Time. Initial Interval is determined by Max Drift between two nodes and Guard Time. It can be calculated by (8). To typical 32768 Hz crystals, their drift between two crystals would not exceed 30 ppm. When Guard Time is set to 100 µs, synchronization interval can be initialized as 3.3 seconds according to the following equation:

\[
\text{Initial} = \frac{\text{Guard Time}}{M_{\text{drift}}}.
\]

When Required Accuracy is set to 100 µs, we can expand the interval by 3.3 times if 32768 Hz crystal is used. In practice we can use a smaller value to avoid the slight fluctuation of drift. Since the expansion is exponential, smaller value will not reduce the performance of Adaptive Compensation.

When synchronization interval expands to some degree, the fluctuation of temperature will become more apparent. Crystal’s drift will also fluctuate. In such case, (1) for calculating drift is not suitable yet. To deal with the influence of temperature, a reboot mechanism based on measurement of temperature is performed.

Using the equipped sensor, nodes can measure the temperature surrounding them periodically and compare it with the temperature which is recorded at last time when node exchanges packet for synchronization. Once nodes detect the fluctuation of temperature exceeding a threshold, it will send a packet to synchronize immediately and reset the synchronizing interval to initial value. Then Adaptive Compensation will start again. Setting the threshold to 2°C, in our experiments, can be tuned to match the crystals temperature sensitivity and the application requirements. Since nodes only need to store the temperature recorded at last time when exchanging packet for synchronization, it would not consume much energy to maintain this data. Also the power cost by sensing is quite small which need not be counted in.

4.2. Synchronization with Adaptive Compensation. To use this Adaptive Compensation method in practice, two parameters are required: Initial Interval for synchronizing and Required Accuracy, or the length of Guard Time. Initial Interval is determined by Max Drift between two nodes and Guard Time. It can be calculated by (8). To typical 32768 Hz crystals, their drift between two crystals would not exceed 30 ppm. When Guard Time is set to 100 µs, synchronization interval can be initialized as 3.3 seconds according to the following equation:

\[
\text{Initial} = \frac{\text{Guard Time}}{M_{\text{drift}}}.
\]

When Required Accuracy is set to 100 µs, we can expand the interval by 3.3 times if 32768 Hz crystal is used. In practice we can use a smaller value to avoid the slight fluctuation of drift. Since the expansion is exponential, smaller value will not reduce the performance of Adaptive Compensation.

When synchronization interval expands to some degree, the fluctuation of temperature will become more apparent. Crystal’s drift will also fluctuate. In such case, (1) for calculating drift is not suitable yet. To deal with the influence of temperature, a reboot mechanism based on measurement of temperature is performed.

Using the equipped sensor, nodes can measure the temperature surrounding them periodically and compare it with the temperature which is recorded at last time when node exchanges packet for synchronization. Once nodes detect the fluctuation of temperature exceeding a threshold, it will send a packet to synchronize immediately and reset the synchronizing interval to initial value. Then Adaptive Compensation will start again. Setting the threshold to 2°C, in our experiments, can be tuned to match the crystals temperature sensitivity and the application requirements. Since nodes only need to store the temperature recorded at last time when exchanging packet for synchronization, it would not consume much energy to maintain this data. Also the power cost by sensing is quite small which need not be counted in.

5. Implementation

We implemented Adaptive Compensation based on OpenWSN’s work. OpenWSN project provides open-source implementations of a complete protocol stack based on Internet of Things standards. It uses IEEE802.15.4e as MAC layer protocol of the stack. Basic time slot communication
and packet-based synchronization defined in IEEE802.15.4e have already been implemented by OpenWSN. Since the synchronization of IEEE802.15.4e did not adopt compensation method yet, we add codes based on OpenWSN’s implementation to make our compensation method work on IEEE802.15.4e.

The hardware of this implementation is called GINA (Figure 4). GINA is a low-cost wireless node containing an MSP430F2618 [14] 16-bit microcontroller and a AT86RF231 [15] radio chip. An external crystal of 32768Hz, with a drift of ±20 ppm, is used to provide a slot length of 10 ms.

When implementing Adaptive Compensation on GINA, four parts are needed. First is the calculation of drift. According to (1), offset and $\Delta t$ are required. offset can be derived just by the synchronization of IEEE802.15.4e implemented by OpenWSN. Through absolute slot number (ASN), nodes can get $\Delta t$ counted by the number of elapsed time slots since last packet-based synchronization. Thus drift can be expressed by how many time slots are elapsed when two clocks move one tick apart.

Second is compensation. Since the drift is calculated by time slots, node can count the number of slots at the beginning of each slot. When the counter hits the value of calculated drift, the synchronized node will lengthen or shorten its current slot length by one tick for compensation. Another part is scheduling synchronizing packets. We can schedule the synchronizing packet with the timer in MSP430F2618. Once the timer fired, an interrupt is triggered to notify nodes to send a packet for resynchronization. The counter of timer will be reset to one time larger value which indicates next synchronizing interval.

The last part is reboot mechanism. There are kinds of sensor equipped on GINA, such as 3-axis gyroscope, 3-axis magnetometer, and temperature sensor. With the temperature sensor, we can detect the fluctuation of temperature and implement reboot mechanism on GINA. TMP20 [16], which is selected to be the temperature sensor of GINA, operates from −55°C to +130°C on a supply voltage of 2.7 V to 5.5 V with a supply current of 4 $\mu$A. The deviation would be ±0.23 when temperature ranges from −10°C to +60°C. To both the temperature accuracy and energy efficiency, TMP20 can totally meet the application requirements.

6. Evaluation

To evaluate the performance of Adaptive Compensation, we did experiments in different scenarios with the implementation. As mentioned before that temperature is the main factor influencing drift, we define three scenarios to evaluate the performance of proposed method: constant temperature, temperature with daily fluctuation, and sudden change in temperature. Those scenarios with different fluctuation of temperature result in diversified performances for Adaptive Compensation. In this section, we describe the settings of those experiments and experimental results in the three scenarios, respectively.

6.1. Experimental Settings. During experiments, slot boundary and synchronizing points are recorded by logic analyzer. Figure 5 shows a snapshot for a simple network of two synchronized nodes, which are running Adaptive Compensation implementation. Node A represents the time parent. The activities of two nodes are indicated by three digital signals:

(i) Synchronization toggles when a resynchronization happened on node B;
(ii) Slot of Node B toggles at each new slot of node B. Each slot is 10 ms long;
(iii) Slot of Node A toggles at each new slot of node A. Each slot is 10 ms long.

Through comparing the slot boundary between nodes A and B, we can record the synchronization error of each slot. Matlab is used for statistics and analysis of the performance of Adaptive Compensation.

The experimental synchronized node has a drift of 12 ppm. Synchronization accuracy of 100 $\mu$s is needed to be maintained for the node. Synchronizing interval is initiated to 8 seconds and it would expand by 2 times when using Adaptive Compensation. The experimental scenarios and results are shown in Figures 6 and 7.

6.2. Experimental Scenarios

6.2.1. Constant Temperature. To indoor environment, the temperature is usually considered as constant. As shown in Figure 6(a), temperature in 24 hours ranges from 25°C to 28°C and varies smoothly. In such environment, the temperature would not influence the drift of crystal. The indoor performance of Adaptive Compensation will be the best in three scenarios.

6.2.2. Temperature with Daily Fluctuation. Unlike indoor case, temperature will become more fluctuant by the influence of complicated outdoor environments. The daily temperature outdoor ranges from 22°C to 29°C as shown in Figure 6(b). Since many factors influence local temperature, like sunlight, shade, and winds, 2°C to 5°C fluctuation of temperature would happen occasionally. According to the result of experiments, those factors only allow
Figure 5: Logic analyzer records the implemented synchronization with gradually improved compensation procedure. Node A is the time parent.

Figure 6: Temperature fluctuation in 24 hours indoor and outdoor and sudden change in temperature.
Figure 7: Synchronization error for three environments: indoor, outdoor, and sudden change in temperature.

resynchronizing interval to go up to few minutes when using improved compensation. The outdoor performance of Adaptive Compensation is discounted comparing with constant temperature scenarios.

6.2.3. Sudden Change in Temperature. In some cases like industrial application, temperature may increase more than 10°C in less than one minute. To evaluate the performance of proposed method in such scenario, a hot-air gun was used to heat up synchronized node every 8 minutes (Figure 6(c)). Because of the great influence on clock drift, frequently packet-based resynchronizations are required to maintain synchronization. In such case, the performance of Adaptive Compensation would regress to the synchronization without using compensation.

6.3. Experimental Result

6.3.1. Constant Temperature. As shown in Figure 7(a), synchronization error increased more slowly after each resynchronization, which can be derived from the slope of fitting lines A and B. Synchronizing points are indicated by blue triangles. The synchronization error from the synchronizing points 255 s to 511 s and 1024 s to 2047 s is fitted by line A and line B. The slope of line B is smaller than line A, which means that the calculated drift at 1024 s is more accurate than the drift calculated at 255 s.

The slope of fitting line corresponds to \( D_r - D_c \) in (4), which expresses the part of drift that was not counteracted by compensation. In Figure 7(a), the slope of fitting line became smaller and smaller since synchronizing interval became longer. In other words, the accuracy of calculated drift has been improved with expanded synchronizing interval.
Synchronization error will still increase at the rate of error is up, it means that the compensated drift is insufficient. The tendency of synchronization error indicated the relationship between $D_r$ and $D_c$. When the tendency of synchronization error is up, it means that the compensated drift is insufficient. Synchronization error will still increase at the rate of $D_r - D_c$. When the tendency is down, it means that the compensated drift is excessive and synchronization error will increase in an opposite rate of $D_r - D_c$.

Temperature fluctuated heavily after 600 s. The tendency of synchronization error was down after 250 s as calculated drift was greater than real drift. The fluctuation of temperature during 700 s to 800 s made real drift bigger than calculated drift and the tendency turned to up. At about 850 s, TMP20 detected the fluctuation of temperature exceeded the threshold of 2°C. Then the synchronized node reset synchronizing interval to 8 seconds and restarted the compensation procedure.

Because of the fluctuation of outdoor temperature, the synchronizing interval using improved compensation can increase to 4 minutes on average and 6 packets are required to maintain the accuracy of 100 μs. If we use packet-based synchronization only, 30 synchronizing packets are required in 4 minutes, and 80% of packets could be eliminated.

### 6.3.3. Sudden Change in Temperature

The performance of proposed compensation method when temperature changed suddenly is shown in Figure 7(c). When the great change of temperature happened on synchronized node, the node would send packet every second, the same rate with temperature sampling, to keep synchronizing, as the periods from 120 s to 340 s and 590 s to 820 s showed. The blue line with wide width from 120 s to 340 s or from 590 s to 820 s consists of hundreds of triangles which are overlapped with each other. Since there are too many triangles (every second there will be one triangle corresponded), they just look like a line in Figure 7(c).

There is no compensation during those periods. After temperature became stabilized, compensation would be started again. Since synchronized node keeps sending packets and no compensation procedure occurred, the performance of Adaptive Compensation regresses to the synchronization without using compensation if reboot mechanism is also used.

### 6.3.4. Number of Packets for Synchronization

Table 1 makes a conclusion of the performance of packet-based synchronization without compensation, fixed compensation, and Adaptive Compensation in those environments. Fixed compensation is a compensation method usually used for extending synchronization interval with limited degree since it only calculates drift one time. The numbers in the table indicate the average of packets used for maintaining 100 μs of synchronization accuracy in 10 minutes. Since the synchronization without compensation needs sending synchronizing packet every 8 seconds (100 μs/12 ppm), 75 packets are required to maintain the accuracy of 100 μs. The Fixed Compensation method only calculated the drift once with the Initial Interval. Using this Fixed Compensation method, the drift of about 9 ppm can be counteracted. The remaining 3 ppm drift needs to be dealt with by packet-based synchronization every 33 seconds (18 packets in 10 minutes). Since the synchronizing intervals are short, Fixed Compensation method or the synchronization without compensation would not be influenced by the temperature indoor and outdoor.

The numbers of packets in 10 minutes indoor and outdoor using Adaptive Compensation are average values according the experiments results mentioned above. For indoor case, Adaptive Compensation needs 9 packets in one hour to maintain accuracy of 100 μs, 1.5 packets in 10 minutes on average. For outdoor case, 6 packets in 4 minutes means 15 packets in 10 minutes on average.

To deal with sudden temperature change, a temperature detective mechanism is necessary, or the synchronization error will fall out Guard Time and lose synchronization. The number of packets when temperature suddenly changed is determined by the rate of temperature sampling. The value of packet number recorded in Table 1 is measured at the rate of sampling every second.

According to Table 1, the synchronizing packets using Adaptive Compensation indoor is much fewer than Fixed Compensation and the synchronization only exchanging packets. To outdoor case, packets using Adaptive Compensation are a little fewer than Fixed Compensation but still much fewer than only packet-based synchronization.

| Methods synchronization               | Number of packets for synchronization |
|--------------------------------------|---------------------------------------|
|                                       | Indoor | Outdoor | Sudden change in temperature |
| Without compensation                 | 75     | 75      | 600                              |
| Fixed compensation                   | 18     | 18      | 600                              |
| Adaptive Compensation                | 1.5    | 15      | 600                              |
When temperature suddenly changed, three methods have the same packet number for synchronizing if they use the same sampling rate.

7. Conclusion

This paper proposed a method called Adaptive Compensation, which improving the accuracy of calculated drift by increasing synchronizing interval gradually. This method is used for avoiding frequently packet-based resynchronization to meet the industrial requirement. The improved compensation method was implemented on GINA platform based on OpenWSN’s work. Compared with the synchronization without using compensation, the experiments done indoor and outdoor showed 98% and 80% resynchronization packets are eliminated. The performance of Adaptive Compensation regressed to the synchronization without using compensation when temperature changed suddenly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Y. Wan for his assistance in experiments and J. He for reviewing this paper.

References

[1] K. Sohraby, D. Minoli, and T. Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, John Wiley & Sons, New York, NY, USA, 2007.
[2] D. Networks, “Technical overview of Time Synchronized Mesh Protocol (TSMP),” Tech. Rep., Dust Networks, 2007.
[3] WirelessHART Specification 75: TDMA Data-Link Layer, HART Communication Foundation Std., Rev. 1.1, hCF SPEC-75, 2008.
[4] ISA, “Isa-100.11a-2009: wireless systems for industrial automation: process control and related applications,” September 2009.
[5] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) Std, IEEE Std 802.15.4e, April 2012.
[6] “Berkeley’s OpenWSN project,” http://openwsn.atlassian.net.
[7] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI ’02), pp.147–163, December 2002.
[8] 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 (SenSys ’03), pp.138–149, November 2003.
[9] M. Maroti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04), pp.39–49, 2004.
[10] T. Bian, R. Vankatesan, and C. Li, “Adaptive time synchronization for wireless sensor networks with self-calibration,” in Proceedings of the IEEE International Conference on Communications (ICC ’09), pp. 1–5, June 2009.
[11] F. Ren, C. Lin, and F. Liu, “Self-correcting time synchronization using reference broadcast in wireless sensor network,” IEEE Wireless Communications, vol. 15, no. 4, pp. 79–85, 2008.
[12] J. R. Vig, “Introduction to quartz frequency standards,” Tech. Rep. SLCET-TR-92-1 (Rev.1), Army Research Laboratory, 1992.
[13] “OpenWSN Hardware: GINA platform,” https://openwsn.atlassian.net/wiki/display/OW/GINA.
[14] MSP430x2xx Family User’s Guide, SLAU144E, 2008.
[15] Low Power 2.4 GHz Transceiver. AT86RF231-ZU, AT86RF231-ZF, 8111CMCU Wireless09/09, 2009.
[16] Low-Power, Analog Out Temperature Sensor, SBOS466, 2009.