Synchronization of Wireless Sensor Networks Using a Modified IEEE 1588 Protocol

Darold Wobschall and Yuan Ma
Esensors, Inc.
Amherst, USA

***Abstract*— A method of precise time synchronization of wireless sensors employing an IEEE 802.15.4 transceiver, and specifically employing the 6LoWPAN protocol, was developed. It uses the IEEE 1588 synchronization standard and the IEEE 1451.5 Smart Transducer Data standard. A Wireless Transducer Interface Module (WTIM) was designed and fabricated. It utilizes the IEEE 802.15.4 transceiver model TI CC2430 which allows access to a hardware sync signal. The difference in timestamps between two WTIMs was measured. The results show that the synchronization precision is better than 10 µs for short synchronization intervals but increases to about 100 µs for longer synchronization intervals (1 sec for crystal accuracies of 50ppm). The method was tested for 6LoWPAN wireless protocol but would apply to other wireless sensors based on the IEEE 802.15.4 protocols.**

Keywords*-1588; 1451; wireless sensor network; synchronization*)

I. Introduction

The timing of sensor data or actuator control in a wireless network often is critical to the data acquisition or control process. The IEEE 1588 standard [1, 2, 3] describes the time synchronization process for wired network nodes, in particular using Ethernet, but does not explicitly address how this might be extended to wireless networks. Wireless sensors are difficult to synchronize because the total transmission time is comparatively long and quite variable, especially if re-transmission and relaying of messages between nodes is involved. Also energy and bandwidth restriction limit the length and frequency of synchronization messages. We utilize the IEEE 1451.0 and 1451.5 smart transducer standards which provide a standard sensor protocol. These standards do not explicitly specify a method of synchronizing time clocks between the different nodes, or Wireless Transducer Interface Modules (WTIMs), of the network. We have combined the IEEE 1451 and 1588 protocols in this research.

Specific features of this implementation are (1) synchronization pulses are derived from a source close to the physical layer of the transceiver without hardware modifications, (2) implementation with the Internet-compatible 6LoWPAN communication protocol and (3) a precisely synchronized real time clock module with IEEE 1451/1588 format fabricated from commercial components. This allowed synchronization to under 100 µs, and under ideal conditions, to under 1 µs.

II. Description of Synchronization Process

A. Wireless Network Block Diagram

A block diagram of a Wireless Transducer Interface Module (WTIM) and an associated gateway or NCAP front end is shown in Fig. 1. The gateway also functions as a wireless router for the 6LoWPAN network. Both consist of a RF transceiver with IEEE 802.15.4 [8] capability, a microcontroller (which is integrated with the receiver for our system) and a clock module (which here is separate).

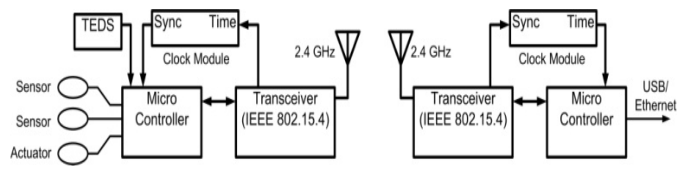

Figure 1. Block Diagram of WTIM (left) and gateway/router or NCAP front end (right)

B. IEEE 802.15.4 Transceiver

The transceiver selected (Fig. 2) is the TI [10] model CC2430 (2.4 GHz) because it is well suited for the synchronization process. In addition to being based on IEEE 802.15.4 protocol [9], two key advantages are access to a hardware synchronization signal and the integration of a microcontroller with the transceiver in the same chip. The time synchronization signal selected is SFD (start frame delimiter), as shown in Fig. 2. It goes high when the initial, required preamble of the message is received (or transmitted). The signal is used as an interrupt on the microcontroller to provide the timestamp provided by the external clock module (Fig. 3). For applications which do not require highly accurate timing, the CC2430 internal MAC timer may be used instead. Because the software for this function is deterministic (non-branching), and crystal controlled, it has little jitter (under 1 µs). The SFD function is similar to the optional beacon.

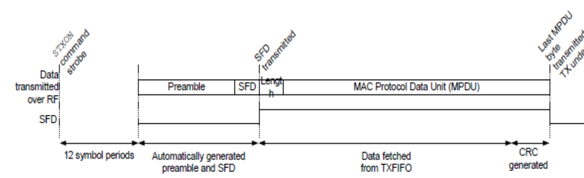

Figure 2. IEEE 802.15.4 Message Timing
-- The SFD functions as the time sync pulse

978-1-4244-5977-3/10/$26.00 ©2010 IEEE

As shown in Fig. 3, there is a propagation delay between the SFD byte transmitted and that received. The transmitter must record the actual time the SFD was transmitted since often there is a delay between the initiation of the transmit process and the time the message is transmitted. Even if the preamble is partially cut during the reception process, the time of the SFD reception is not affected. Thus the falling edges of the SFD are the desired sync signals for the transmitter and receiver.

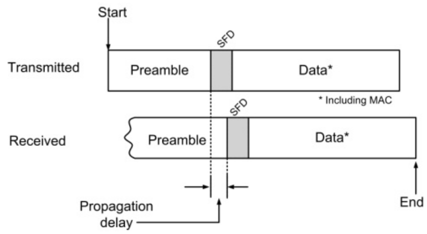

Figure 3. Propagation Delay of Sync (SFD) Signal

C. Synchronized Clock Module

The synchronization clock module (Fig. 4) is a small, fast microcomputer [17] with a precision oscillator crystal. It is a real time clock with the format and features needed for the synchronization. There is one on each WTIM and also on the gateway (NCAP). The crystal may have a standard accuracy (20 to 100 ppm), a precision (1 to 10 ppm) crystal is preferred. A timer within the microcomputer has a resolution equal to the oscillator period (0.0625 µs or better). Since our highest target synchronization precision is 1 µs (with rapid updating), the timer resolution is a better than required for all of our intended applications.

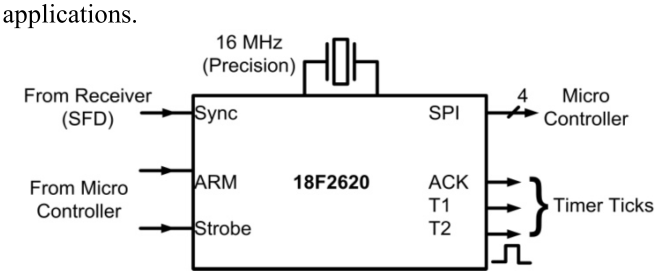

Figure 4. Synchronized Clock Module

-- This functions as a real time clock which can be synchronized and frequency compensated

The time format specified in the IEEE 1451.0 standard [4] is the 64-bit TAI format which is the same as PTP except without the leap seconds (see below for more details). The upper 32 bits is the number of seconds since 1900 (Epoch). The lower 32 bits is the number of nanoseconds. It is based on Greenwich Mean Time, not local time. TAI is optional for IEEE 1588 and it is necessary to convert between it and the NTP time format used on the Internet (1900 Epoch with leap seconds and binary fractions of a second). Normally this is done outside the NCAP (or WTIM).

The precision required of the timestamp, and thus the synchronization precision, depends very much on the application. For many wireless applications, including most industrial control and monitoring processes, a timestamp precision with an error under 1 ms, or perhaps 0.1 ms, is sufficient. By contrast wired (Ethernet) networks with IEEE 1588 protocols are able to provide much closer synchronization.

Synchronization of wireless sensors with accuracy less than 100 µs can be a challenge because of the round-trip message times are of the order of 100 ms and have high variability depending on message length and traffic. However our time sync algorithms are not influenced by the message length and traffic. The RF message itself is about 2-5 ms but is precisely timed, that is, has little jitter. Therefore the errors in our time sync processing are caused primarily by crystal jitter. In any case, our focus here is on reliability and low wireless power requirements, in particular for manufacturing and monitoring applications, rather than on high precision applications which may require frequent updates.

B. Gateway/Router/NCAP

This unit has three functions: Wireless router, wireless gateway and NCAP (Network Capable Application Processor), as shown in Fig. 5. The gateway connects the wireless devices to the Internet and functions as an NCAP which implements the IEEE 1451 protocols. The microcontroller implements the IEEE 1451.5 NCAP format and communications over the Internet using HTTP protocol [7].

Another function of the gateway is the router for the 6LoWPAN network. It manages the traffic between the various nodes of the system.

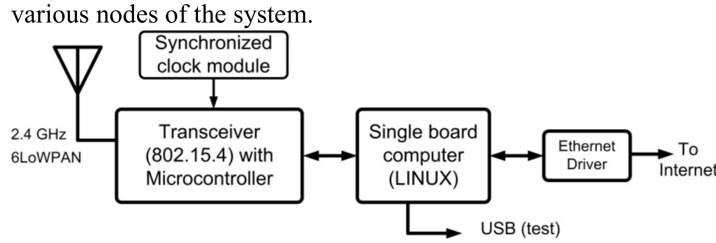

Figure 5. Block Diagram of Gateway with Router
-- Functions as the wireless network router, IEEE 1451 NCAP and Internet Driver

Still another function of the gateway will be to implement the IEEE 1588 precision clock synchronization protocol over the Ethernet so that other nodes employing 1588 clocks are compatible with the wireless devices

C. 6LoWPAN Stack Refinements

The 6LoWPAN Low power Wireless Personal Area Network [17] was designed to enable IPv6 Internet-compatible datagrams over the lower power and bandwidth IEEE 802.15.4 radio. In effect the protocol subdivides the verbose IPv6 messages for efficient transmission over the limited bandwidth and message size radio. The protocol is well suited for sensor data being interfaced with the Internet. While the protocol adds complexity on the RF communication side, it simplifies the process from the user point of view.

The 6LoWPAN stack [18], although under development for several years by various groups, is not yet fully developed. Many 6LoWPAN stacks available from proprietary sources are unsuitable for implementing the IEEE 1588 synchronization process but we were able to make the software stack from Sensinode [17] operational for our WTIM.

Because the synchronization is implemented at the MAC layer, it is not specific to the 6LoWPAN stack and thus any network based on IEEE 802.15.4, such as Zigbee, can use the same method.

D. Modified IEEE 1588 Time Synchronization Protocol

The IEEE 1588 Precision Time Synchronization Protocol [1] provides a standard method of synchronizing clocks of the nodes of networks. The standard was developed for wired networks, specifically Ethernet. Our goal is use the IEEE 1588 core concepts with wireless networks but realize that it cannot be done without modifications for a practical implementation. The problems of directly applying the wired standard without modification to wireless are:

- The standard IEEE 1588 synchronization message (166 bytes) is too long for IEEE 802.15.4 (128 byte limit) which is used for 6LoWPAN (and Zigbee). It must be shortened or fragmented.
- The data rate of wireless is far less than wired (e.g. 250 kbits/sec for IEEE 802.15.4 and 100 Mbits/sec for Ethernet). At least buffering and retiming is required.
- The computer resources for standard, wired IEEE 1588 require a relatively large amount of processing power (and storage) which is inadvisable for battery-operated wireless systems

These problems are discussed in the paper "Precision Time Synchronization using Wireless Sensor Networks" by Cho et al [19] and other papers describing the HRTS, TPNS and RBS protocols [12 - 16].

The protocol we use is shown in Fig. 6. It is similar to the standard IEEE 1588 diagram except that the follow-up transmissions are optional since there is a fixed time between the sync message initialization time T1' and T1, as well as T3' and T3. It also can be compared with Fig. 6 of Cho et al [19] where several fragmented messages are sent wirelessly rather a single message which we send

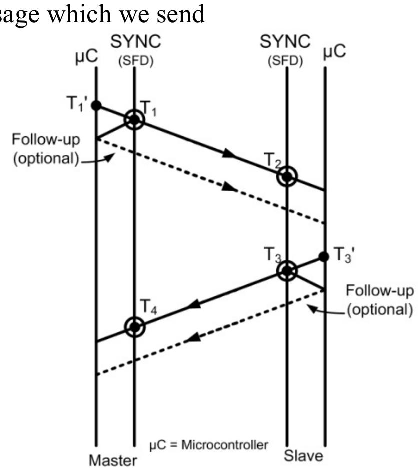

Figure 6. Time Synchronization between Master (gateway/NCAP) and Slave (WTIM)

The primary differences between the wired and wireless versions of IEEE 1588 are:

- The on-the-air sync message is abbreviated (required for IEEE 802.15.4 unless message is fragmented, requiring more power).
- Follow-up messages are normally not used for the modified version (but are with the full-format version) because the time between the sync message initialization and sync time are fixed, known quantities (saves power and precision not affected).

We believe that the on-the-air message format is not critical to whether the protocol can be properly termed "IEEE 1588 protocol" but rather that {1} the procedure must adhere to the core method of 1588 which involves determining and compensating the clocks offset and transmission delay times, {2} there must be conformity to the 1588 format on the network (Ethernet) side, and, of course, {3} it must provide precision clock synchronization. It may be possible to extend the IEEE 1588 standard to include wireless in the future. A version of the wireless IEEE 1588 protocol which uses the full IEEE message (fragmented transmission) and the follow-up messages (Fig. 6) is available.

E. IEEE 1451.5 Protocols and Formats

The IEEE 1451.5 (Dot 5) smart transducer was partially implemented based on the IEEE 1451.0 (Dot 0) standard [4]. Dot 0 specifies the TEDS (Transducer Electronic Data Sheet, located in the WTIM) sensor, commands and data structure in detail, but without reference to the specific physical layer of the network or digital interface. Dot 5 [5] adds the physical layer for several (currently 4) wireless sensor networks, including 6LoWPAN. The 1451 protocol is implemented at the application layer and does not directly affect the 1588 synchronization process which occurs at the MAC layer.

III. CLOCK PRECISION AND POWER TRADEOFFS

Because the frequencies of the clock crystal on the WTIMs or slaves are not exactly equal to that of the NCAP or master, their clocks will gradually drift apart after being synchronized. As shown in Fig. 7, the measured frequency of a slave increases or decreases linearly with the time after the synchronization pulse is applied, depending on whether its frequency is higher or lower than the master. A lower slope occurs for more accurate or better matched crystals. Note that the intercept and jitter are under 1 μs suggesting that the synchronization process itself is quite precise and that with frequent updates a precision better than 1 μs may be achieved. Cho [19], using similar hardware, reported a precision better 0.1 μs with rapid updates and extensive period averaging. Since the drift is predictable after the first sync interval, software method for correcting the internal clock frequency can result in a significant improvement of the time clock precision (not shown here) whether standard or precision crystals are used.

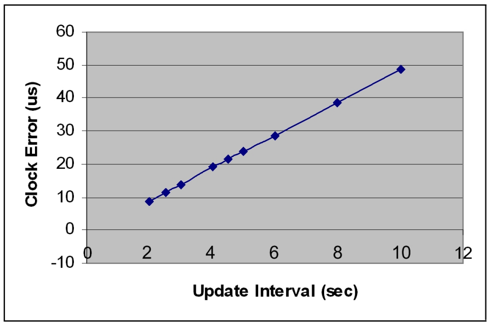

Figure 7. Measured Timing Error (µs) vs. Sync Update Interval --Clocks on WTIMs drift apart after sync due to small crystal frequency inaccuracies

While the clock precision can be much improved by more frequent sync updating, it comes at the cost of increased power consumption of the wireless devices. This is particularly true for WTIMs normally in a sleep mode and which only transmit when necessary. An example of the average node power vs clock precision is shown in Fig. 8. Note that the power required, and therefore battery life, varies by several orders of magnitude. Frequent transmissions also clutter the band which may have significant effects on other wireless traffic. It is therefore important that the clock precision not be over-specified and to recognize that precisions better than 0.1 to 1 ms are not often needed.

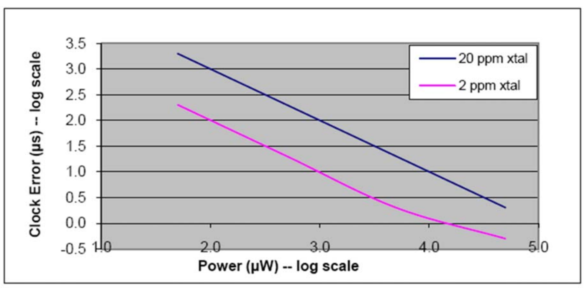

Figure 8. Timing precision vs Wireless Power Consumption -- Higher and lower precision crystals without software compensation used

A comparison of the clock errors vs time after reset for several WTIMs is shown in Fig. 9.

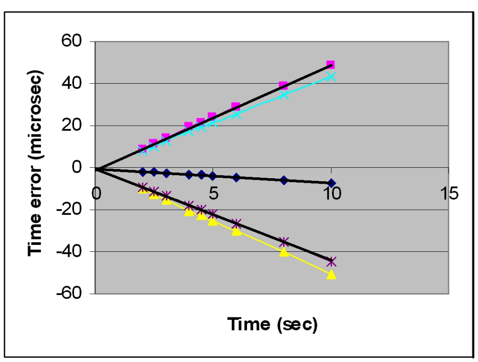

Figure 9. Measured Timing Error (µs) vs. Time after Synchronization for Several WTIMS.

IV. CONCLUSIONS

A wireless sensor system based on the IEEE 802.15.4 protocol and the TI CC2430 chip was synchronized using a modified version of IEEE 1588 protocol. By using the SFD signal on the chip, synchronization to under 1 µs was achieved under good conditions with rapid updating. With slower updating, much less power was needed resulting in both long battery life and a still acceptable (1 ms) precision. Further work is being done to establish the precision with some RF interference and less ideal conditions.

REFERENCES

[1] Description of IEEE 1588 standard: http://www.ieee1588.com/
[2] Agilent and 1588: http://cp.literature.agilent.com/litweb/pdf/5989-6147EN.pdf
[3] National Instruments and 1588: http://zone.ni.com/devzone/cda/pub/p/id/130
[4] IEEE 1451.0-2007, Standard for a Smart Transducer Interface for Sensors and Actuators–Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.
[5] IEEE 1451.5-2007, Standard for a Smart Transducer Interface for Sensors and Actuators–Wireless Communication and Transducer Electronic Data Sheet (TEDS) Formats.
[6] IEEE 1588-2008, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.
[7] IEEE 1451: K. Lee and M. Reichardt "Open Standards for Homeland Security Sensor Networks," IEEE Instrumentation and Measurement Magazine, Dec. 2005
[8] D. Wobschall, "Network Sensor Monitoring using the Universal IEEE 1451 Standard," IEEE I and M, Vol. 11, p 18 (2008).
[9] IEEE 802.15.4 http://www.amazon.com/802-15-4-Low-Rate-Wireless-Personal-Networks/dp/0738135577
[10] TI. CC2430 Datasheet: A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee(TM) (Rev. F). 2007
[11] FREERTOS: http://www.freertos.org
[12] Jeremy, E., G. Lewis, and E. Deborah, Fine-grained network time synchronization using reference broadcasts, in Proceedings of the 5th symposium on Operating systems design and implementation. 2002, ACM: Boston, Massachusetts
[13] Mikl, et al., The flooding time synchronization protocol, in Proceedings of the 2nd international conference on Embedded networked sensor systems. 2004, ACM: Baltimore, MD, USA
[14] Ping, S. (2003) Delay Measurement Time Synchronization for Wireless Sensor Networks
[15] Saurabh Ganeriwal, R.K., Mani B. Srivastava, Timing-sync protocol for sensor networks, in Proceedings of the 1st international conference on Embedded networked sensor systems. 2003, ACM: Los Angeles, California, USA
[16] Han, H.D.a.R., Tsync: A lightweight bidirectional time synchronization service for wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review, 1994. 8: p. 125--139
[17] 6LoWPAN stack: Sensinode (Oulu, Finland), NanoStack Manual v1.0.0, http://www.sensinode.com/
[18] Wireless time sync reference, H. Cho, S. Son and Y. Baek "Implementation of a Precision Time Protocol over a Low Rate Wireless Personal Area Networks" IEEE Computer Systems Architecture Conference, 2008. ACSAC 2008. 13th Asia-Pacific