Synchronization of Wireless Sensor Networks Utilizing Broadcast Signal Time Stamps

Hans-Martin Tröger, Florian Schmittner, Thorsten Nowak, Joerg Robert, Albert Heuberger
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Institute of Information Technology (Communication Electronics), 91058 Erlangen, Germany

***Abstract*—The synchronization of wireless sensor networks is a crucial task, especially for wide area networks. In case of many applications state-of-the-art techniques such as GPS or network based timing protocols are not suitable. In this paper we present a new concept that allows for the efficient synchronization of sensor networks using the time stamps of so-called Signals of Opportunity. We explain how these time stamps can be examined on the example of Digital Audio Broadcasting and show measurement results on the achievable performance.**

***Index Terms*—Synchronization, Time, DAB, OFDM**

I. Motivation

Bats are important indicators for a stable natural environment. However, little is known about their social behavior. Within a research project of the Deutsche Forschungsgemeinschaft (DFG) the department of information technology of the Friedrich-Alexander Universität Erlangen-Nürnberg is developing a tracking system for measuring the flight trajectories of bats within their natural environment. For this purpose the bats are equipped with ultra-light weight active transmitter tags which emit a special waveform with a rate of about 10 messages per second [1]. These signals are then received by a network of stationary base-station receivers. With the data of several base-stations the exact bat position can be estimated with the help of received field strength measurements (RSSI). As the base-stations may operate in areas without sufficient network connection, they locally store the RSSI measurements equipped with additional time-stamps. The actual flight trajectories are then calculated offline after the measurement campaign. This requires a unique assignment of the received bat signals. However, for longer observation times the clocks of the base-stations diverge from each other. If the time error reaches the transmission period of the transmitter tags, i.e. $100\,\mathrm{ms}$, assignment problems of the measured RSSI values occur. Consequently, the relative timing error of the base-stations must be significantly lower than the time between two bat signals, which requires time synchronization between the base-stations. The field of time synchronization of wireless sensor networks (WSN) is a well known problem in the literature [2, 3, 4]. However, most of the proposed algorithms are difficult to implement, especially in case of large WSN without sufficient network connection. A state-of-the-art solution is the use of GPS. However, this method is not suitable for our network, as it denies the operation within dense forests and in the entrance areas of caves. Further, also bidirectional network timing protocols such as NTP or PTP cannot be used without a suitable network connection. The common way of using time-stamps from a long-wave transmitter station is also not a suitable option, as it requires additional hardware, e.g. a magnetic antenna.

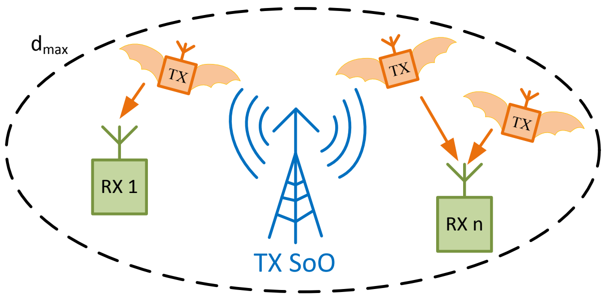

Figure 1. System structure of a distributed WSN synchronized with Signals of Opportunity

For this reasons we are focusing on so-called Signals of Opportunity (SoO) as a source of synchronization information for our base-station network. These signals can be received using our current base-station hardware. Furthermore, modern broadcast signals like European Digital Video Broadcasting Terrestrial (DVB-T) or Digital Audio Broadcasting (DAB) are usually emitted with a high transmission power of several kW leading to excellent signal-to-noise ratios at the receiver position [5, 6]. Most modern broadcast signals furthermore contain time information for the receiver synchronization. The broadcast emitters, which insert the time-stamps into the broadcast signals, are usually GPS synchronized.

In this paper we explain how the time-stamps can be extracted from the broadcast signal with low complexity using DAB signals. Further, we explain how the time information can be used to adjust the local clocks. Finally, we show some measurement results and discuss the achievable performance.

II. System Description

Figure 1 shows the system structure of our bats tracking system. Our application requires a long time observation of the bats with just a simple presence detection. The active transmitters (TX) mounted on the back of the bats emit a special waveform, which is sequentially emitted on $868\,\mathrm{MHz}$ and $2.4\,\mathrm{GHz}$ with a repetition rate of about $10\,\mathrm{Hz}$ [1]. The base-

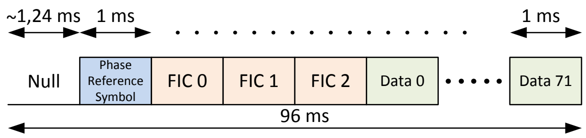

Figure 2. Signal structure of a DAB frame

stations (RX) are constantly receiving on these frequencies and correlate the received data against a pre-defined sequence. If a bat is detected, the current time, the bat ID and the RSSI value are stored locally. At the end of the measurements campaign all RX are collected and the measurement data is merged within a database. Afterwards the estimated flight trajectories can be estimated. Problems occur if the time-stamps of the measurements do not match with each other. This occurs if no time synchronization mechanism is applied to the base-stations. Due the series spill of the reference oscillators the fundamental time increment of each base station is slightly different. After several hours of operation, the local clocks diverge in the order of a second from each other. This time offset is already to high for unique identification of the bat signals within the base-station network.

For this reason we are using SoO to synchronize the base station-receivers. Each modern communication or broadcast signal contains time-stamps required for the signal service. The time synchronization of the RX is achieved by a periodic observation of the SoO. This can be realized using the same hardware that we use for the RSSI measurements. Occasionally the RX tune from the RSSI measurement frequency to the SoO frequency. With an ultra efficient signal decoder the time-stamps are extracted from the SoO data stream. Afterwards, the local clocks are adjusted using this time information. For this approach the maximum dimension of the receiver network is only limited by the quasi error-free reception conditions of the SoO (cf. d_{max} Figure 1). With ideal conditions this can be up to $50\,\mathrm{km}$ or even larger in single-frequency-networks.

III. DAB Signal and Time Stamp Extraction

For the bats localization system we are mainly focusing on the European Digital Audio Broadcasting (DAB) [5] as a source of synchronization. This signal is based on an OFDM waveform with an overall bandwidth of about $1.535\,\mathrm{MHz}$. In Germany the signal is usually emitted in the $200\,\mathrm{MHz}$ frequency band. Figure 2 shows the signal structure of the DAB frame.

In the commonly used DAB transmission mode I a frame consists of 76 OFDM symbols plus an additional transmission break, the so-called Null Symbol. The first information containing symbol is the Phase Reference Symbol (PRS), which is totally known to the receiver. The following OFDM symbols are modulated differentially to this symbol. The next three OFDM symbols transmitting the data of the Fast Information Channel (FIC). This FIC contains fundamental information for the RX. Examples are information about the radio program within the data stream, characterization of program type and the current date and time with millisecond accuracy. The remaining 72 data symbols provide the payload data, e.g. the audio data.

Our bats RX is based on an ARM A-9 dual-core processor, which does not have the processing capability for a continuous reception of the DAB signals. However, this is not required, till all information required data is transmitted within the first 4 symbols of the DAB frame. At first we use an energy detector to find the start of the DAB frame. The data around the Null Symbol is than correlated with the time domain signal of the PRS to reach the correct symbol timing with sample accuracy. After the PRS have been found once, the exact symbol timing is known by counting samples. From now on, only the samples of the PRS and the 3 FIC symbols are processed and the residual samples are neglected. The information of the PRS can also be used for the estimation of the carrier frequency offset and the sampling clock offset [7, 8]. The data content of the FIC symbols is demodulated, the bits are hard decided, and the convolutional code is then decoded by means of a Viterbi decoder to get the data of the FIC. The FIC data is structured in different Fast Information Groups (FIG). The FIG $0/10$ contains the current data and time in the Modified Julian Date format with a resolution of milliseconds. The time stamp represents the transmission time of the start of the PRS. This time stamp can be accessed using a parser within our embedded Linux operating system. The initial Linux system time is replaced by the first available DAB time after system startup. Afterwards the local clock is adjusted using the $adjtime()$ algorithm from <sys/time.h> Linux clock library [9].

IV. Measurement Results

Figure 3 shows the hardware platform of our receiver node. The demodulation and decoding algorithms have been implemented in c++ using our data-stream framework “dfc++”[10]. The board is capable of processing the required data of the incoming DAB signal in software. It extracts the time information provided by the DAB signal and uses this information to adjust its local software clock.

The dashed line in Figure 4 shows the free running local time of one sensor node compared to the reference time provided by the DAB signal. The oscillators is of the type DSC1001DI1 with a frequency of $33.33\,\mathrm{MHz}$. It is obvious that the local oscillators has a frequency offset that leads to a linearly rising time offset. The clock offset is already $5\,\mathrm{s}$ after only 15 hours of operation. When the synchronization is turned on, the time error can be kept at very low levels as shown in Figure 4 (solid line). This proves that distributed clocks can be synchronized utilizing DAB time-stamps.

If we take a closer look to the synchronized case in Figure 5 we can see that the local time offset compared to the received DAB time has a statistical mean offset of $279\,\mu\mathrm{s}$ with a a standard deviation of about $1.28\,\mathrm{ms}$. This standard deviation matches the expectation quite well, till the

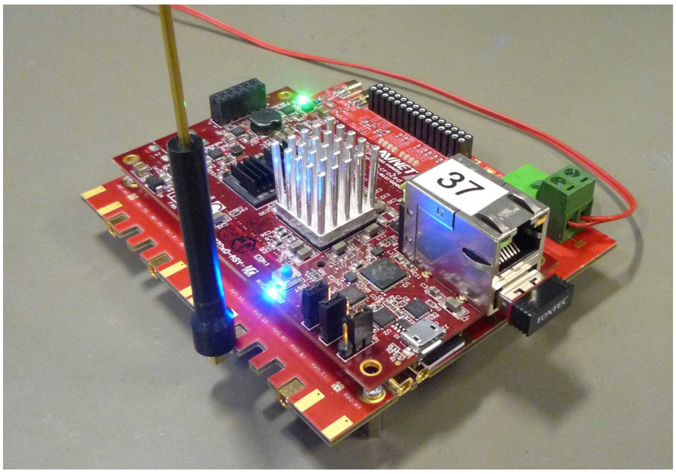

Figure 3. Bats receiver consisting of a self-designed receiver board and a commercial processing board.

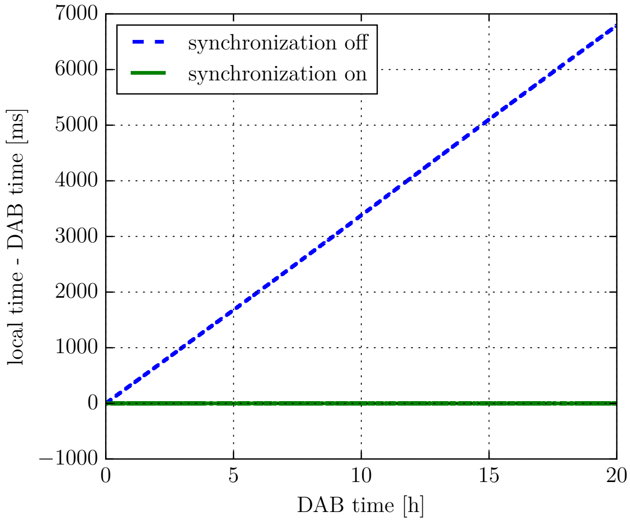

Figure 4. Local time of one sensor node compared to the DAB time

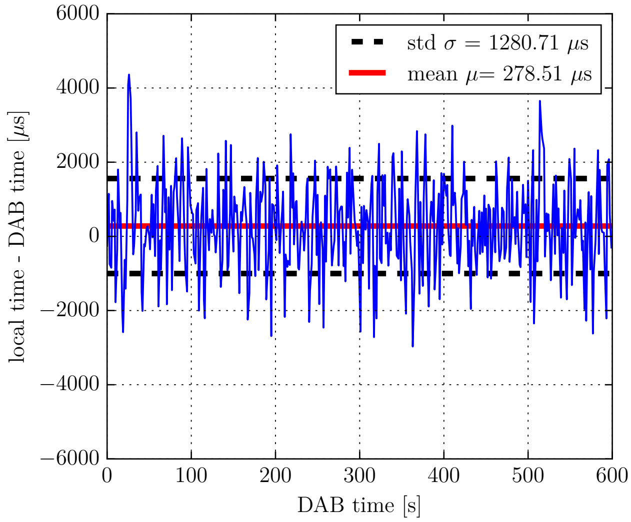

Figure 5. Local time offset in the synchronized case in reference to the DAB time

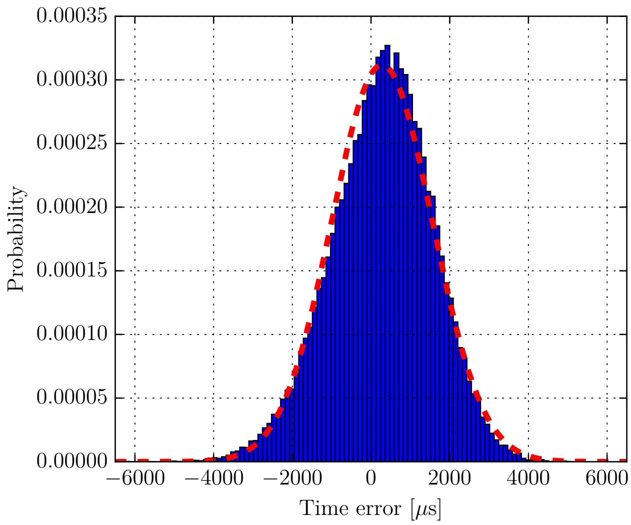

Figure 6. Normalized histogram of the time offset using the time synchronization

time stamp granularity in the DAB signal is in milliseconds. Further, there is a non-deterministic processing delay due to the Linux operating system.

Figure 6 shows the measured histogram of the time offset with synchronization. The measurement results nearly follow a Gaussian distribution, tested by a curve fitting shown as the red dotted line. This tells us, that our measurements follow a stationary process over the total measurement interval.

With the help of the auto-correlation function (ACF), we can test if our measurement results are white or uncorrelated. If we could reach a white auto-correlation, the noise in the measurements would be dominated by a memory-less noise source like additive channel noise. Usually, the oscillators phase noise is suffering from higher order noise processes like flicker frequency and random run. Using the disciplination of our local clock, the influence all these processes can attenuated til the white Gaussian channel noise becomes the dominating noise source. Figure 7 shows the ACF. There is a covariance between the measurements. However, the correlation besides the main peak is low.

The measurement results show the maximum accessible performance of the proposed concept. These measurements assume that every time stamp in the DAB signal (rate of approx. 1.45 s) is used for the adjustment of the local clocks. However, in order to reduce the required processing power we can reduce the number of used time-stamps by using a longer interval between two synchronizations. In the following we analyze the impact of the clock adjustment interval on the accessible accuracy.

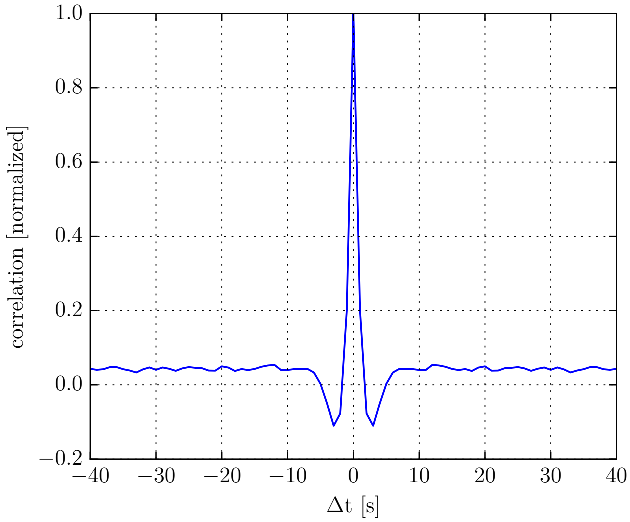

Figure 7. Auto-correlation function of the time error measurements

Table I
ACCESSIBLE ACCURACY OF THE TIME SYNCHRONIZATION UTILIZING DAB TIME-STAMPS WITH DIFFERENT UPDATE INTERVALS

Adjustment interval	10 sec	30 sec	1 min	5 min
Mean µ [ms]	0.8	2.0	3.6	17.2
Standard deviation σ [ms]	1.47	1.8	2.1	7.3
MTIE [ms]	6.8	8.68	11.1	34.0
Adjustment interval	**10 min**	**15 min**	**30 min**	
Mean µ [ms]	33.4	52.4	103.9	
Standard deviation σ [ms]	13.6	20.7	41.4	
MTIE [ms]	61.0	91.8	180.3	

Table I shows the measurement results for different adjustment intervals. For each measurement we run an initial calibration using each time stamp within the DAB signal for clock adjustment for about 60 s. Afterwards, the update period is reduced. Further, we calculate the maximum time interval error ($MTIE$) which gives information about the peak to peak time error within the measurement interval and is a metric for the worst case performance of the system. It is defined by [11, 12]. For a limited measurement interval it follows:

$$MTIE = \max_{0\leq n\leq N}(TE(n)) - \min_{0\leq n\leq N}(TE(n)) \tag{1}$$

where $TE(n)$ is one time error measurement within the total number of N measurements. The TE is the time difference between the disciplined clock and the reference clock. In our case the disciplined clock is the Linux system time and the reference clock is the time from DAB. The measurement results show a steadily rising standard deviation of the time for increased adjustment intervals. However, the loop controller is not ideal, which results in the rising mean offset. A more sophisticated algorithm, which could adjust the fundamental time increment like the $adjtimex()$ from <sys/timex.h> library, could reach significantly better results [13]. For this reason, the measurement results can be understood as the accessible worst-case-performance using a simple proportional loop controller.

V. Conclusion

In this paper we propose a new concept for the wireless synchronization of sensor nodes using Signals of Opportunity (SoO). With a simplified OFDM receiver the time-stamps of the chosen SoO are extracted. This time information can be used to adjust the local time increment. With our demonstrator we are able to shows that such “light-weight” receiver can be implemented on an embedded sensor node with limited processing power. Measurements prove the concept and indicate an achievable timing accuracy of about 1.28 ms. The expected performance reduces with longer update intervals. However, update intervals in the order of one minute are sufficient to keep a standard deviation of about 2.1 ms. Thus, the measurements show that a combined localization and synchronization with only one receiver path is possible.

Acknowledgment

This contribution was supported by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology as a part of the Bavarian project ”Leistungszentrum Elektroniksysteme (LZE)”. More information can be found under http://www.lze.bayern.

References

[1] T. Nowak, M. Hierold, A. Koelpin, M. Hartmann, H.-M. Troger, and J. Thielecke, “System and signal design for an energy-efficient multi-frequency localization system,” in *2014 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, Jan. 2014, pp. 55–57.

[2] D. Djenouri and M. Bagaa, “Synchronization Protocols and Implementation Issues in Wireless Sensor Networks: A Review,” *IEEE Systems Journal*, vol. 10, no. 2, pp. 617–627, Jun. 2016.

[3] Y. C. Wu, Q. Chaudhari, and E. Serpedin, “Clock Synchronization of Wireless Sensor Networks,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[5] European Standard, “ETSI EN 300 401 - Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers, V1.4.1,” 2006.

[6] European Standard, “ETSI EN 300 744 - Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television, V1.6.1,” 2009.

[7] M. Speth, S. Fechtel, G. Fock, and H. Meyr, “Optimum receiver design for OFDM-based broadband transmission II. A case study,” *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 571–578, 2001.

[8] H.-M. Troeger, M. Hartmann, J. Robert, J. Thielecke, and A. Heuberger, “Improved OFDM Frequency Syntonization using Signal Decision and Re-Modulation,” Honolulu, HI, USA, Apr. 2015.

[9] “adjtime(3) - Linux manual page.” [Online]. Available: http://man7.org/linux/man-pages/man3/adjtime.3.html

[10] D. Soller, T. Jaumann, G. Kilian, J. Robert, and A. Heuberger, “DFC++ Processing Framework Concept,” *Journal of Signal Processing Systems*, pp. 1–10, Aug. 2016. [Online]. Available: http://link.springer.com/article/10.1007/s11265-016-1174-x

[11] S. Bregni, “Measurement of maximum time interval error for telecommunications clock stability characterization,” *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 5, pp. 900–906, Oct. 1996, 00034.

[12] N. I. of Standards and Technology, Ed., *NIST Special Publication 1065 - Handbook of Frequency Stability Analysis*, 00138.

[13] “adjtimex(2) - Linux manual page.” [Online]. Available: http://man7.org/linux/man-pages/man2/adjtimex.2.html