Proceedings of the 24th International Conference on Automation and
Computing, Newcastle University, Newcastle upon Tyne, UK, 6-7
September 2018

Implementation of Timestamped Pulse-Coupled Oscillators in IEEE 802.15.4 Networks

Yan Zong, Kexin Luo, Xuewu Dai, Zhiwei Gao
Department of Mathematics, Physics and Electrical Engineering
Northumbria University
Newcastle upon Tyne, United Kingdom
{yan.zong, kexin.luo, xuewu.dai, zhiwei.gao}@northumbria.ac.uk

***Abstract*—Time Synchronization (TS) is a key enabling technology of mission-critical Wireless Sensor Networks (WSNs) to provide a common timescale for distributed sensor nodes. Inspired by synchronous flashing of fireflies, a bio-inspired model, Pulse-Coupled Oscillators (PCO), has been intensively studied. The most studies on PCOs are theoretical work, and the assumption is given that oscillators broadcast and receive the *Pulse*s simultaneously when synchronization of a network is achieved. This is not true when it comes to any real-world environments. From the viewpoint of WSNs, the clock of a sensor node driven by crystal oscillators can be modelled as an oscillator, and *Pulse* firing can be implemented by transmitting a packet. However, the concurrent transmission of *Pulse* packets is impossible due to the packet collision in the single wireless channel. To avoid this issue in WSNs, this paper adopts a *desynchronization* mechanism, in which the *Pulse* packets are transmitted to the wireless channel in a uniformly distributed fashion and in accordance with standard IEEE 802.15.4. A hardware testbed is developed to implement the *desynchronized* pulse-coupled oscillators, and it can also be extended to the large-scale wireless sensor networks.**

***Index Terms*—time synchronization, pulse-coupled oscillators, *desynchronization*, IEEE 802.15.4, wireless sensor networks**

I. Introduction

In many Wireless Sensor Networks (WSNs) applications, high precision common sense of timing among sensor nodes is required to enable the distributed nodes to work co-operatively to monitor events or objects in time-sensitive wireless network applications, such as collaborative condition monitoring, intruder detection and time-of-flight location [7].

In the embedded systems, the discrete clock consists of a crystal oscillator running at a specified frequency, and a register *counter* that counts the number of clock ticks generated by a crystal oscillator. As for the low-cost embedded systems, the local clock, which is commonly driven by a crystal oscillator, is subject to variation in phase resulting from the manufacturing tolerance. In addition, owing to the environmental conditions, clock frequency may also be drifting. Therefore, the goal of Time Synchronization (TS) is to keep the phase and frequency drift of these sensor node clocks as close to zero as possible. Finally, once the time synchronization is achieved, all the clocks of sensor nodes in a network attain the same timescale.

Due to its foundation and significance, considerable attention has been devoted to the synchronization in the mathematics and physics community, where it is applied to networked oscillators consisting of a set of oscillators whose phases are pair-wise coupled. The phase-coupling in the most of theoretical work is analogue and smooth. It is not suitable for the existing discrete wireless sensor networks. Inspired by the fireflies' behavior, the Pulse-Coupled Oscillators (PCO), being a typical model of networked oscillators, was proposed by [4]. The episodic, pulse-like and discrete phase-coupling (also named with pulse-coupling) of PCO is suitable in the existing WSNs [7].

A. Related Work

An oscillator of the classical PCO works either in free-running mode or interactive mode. In the free-running mode, the clock state, which is denoted by P, rises toward a threshold value. Once the clock state reaches the threshold, the oscillator fires, namely, a *Pulse* is broadcasted and P is reset, after which the cycle repeats.

The pulse-coupled oscillators can be modelled as packet-exchange time synchronization in WSNs, where the *Pulse* firing in classical PCO is equivalent to the packet transmission, and the clocks in the wireless sensor nodes are equivalent to oscillators in the PCO's free-running mode. Thus, the term *oscillator* is used when describing the classical PCO, the term *node* is adopted when describing the implementation of PCO in WSNs, and the wireless *SYNC* packet is used to describe the *Pulse* of PCO model [7].

In the PCO's interactive mode, the clock state evolves as mentioned in the free-running mode. In addition, due to the excitatory coupling between coupled oscillators, the clock state P is pulled up when a *Pulse* is received from another oscillator. Finally, all the oscillators of a network will broadcast *Pulse*s simultaneously when the synchronization is achieved.

Due to its inherent scalability and simplicity benefiting to be applied to the large-scale WSNs, PCO has also attracted a lot of attention over the years. However, when the synchronization of the sensor network is achieved, a large number of synchronized sensor nodes will transmit numerous *SYNC* packets to the single wireless channel at the same time. As a result, the *SYNC* packets from all the synchronized

Yan Zong gratefully acknowledges financial support from the University of Northumbria at Newcastle *via* a postgraduate research studentship.

sensor nodes will interfere with each other (known as packet collision) and no *SYNC* packets can be successfully received. To solve this issue, the *desynchronization* (DESYNC) was proposed by [2] to ensure the *SYNC* packets be transmitted in a uniformly distributed fashion.

[2] implemented the *desynchronization* in a single-hop wireless sensor network. When DESYNC is achieved in the wireless network, N sensor nodes are able to transmit the *SYNC* packets during one time synchronization cycle. The existing literatures [1] and [2] provided no solution for separating the *SYNC* packets from the data transmitted on the same wireless channel. Even though [3] proposed and implemented a scheduling protocol that allows natural separation of the *SYNC* packets from the data transmitted on the same channel, it still lacks the compatibility between the *desynchronization* and the IEEE 802.15.4 superframe.

B. Contribution and Paper Organization

In this paper, the *desynchronized* pulse-coupled oscillators model, which enables the extension of PCO in the large-scale WSNs and the compatibility with existing IEEE 802.15.4, is implemented on a hardware testbed to fully evaluate its performance in a real-world context. The rest of this paper is as follows: Section 2 presents the network architecture and functions of wireless sensor nodes, followed by the *desynchronized* PCO protocol in section 3. Next, the hardware platform in Section 4 is introduced, and the software development is given in Section 5. Section 6 gives proof-of-concept tests and performance evaluation. Finally, conclusions are presented in Section 7.

II. Network Architecture

The cluster-tree is a widely used network topology in most WSNs applications. Fig. 1 exemplifies a typical cluster-tree network which is the topology considered in this paper. The network is identified by the unique *root node*, equipped with a Global Positioning System (GPS) clock. The *root node* works as a pacemaker in the WSNs, it can only transmits the *SYNC* packet to the wireless channel to provide *reference time* to other senor nodes, rather than receiving the *SYNC* from channel to correct its own clock. Each *internal node* is an node of a tree that has child nodes, and the *leaf node* is the node without any child nodes. As the Full Function Device (FFD) in the WSNs, both *internal node* and *leaf node* are able to send/receive the *SYNC* packet from/to the wireless channel. Throughout this paper, the *reference clock* is referred to as master clock, and the *root node* and *internal* (and *leaf*) *node* are called master node and sensor node respectively.

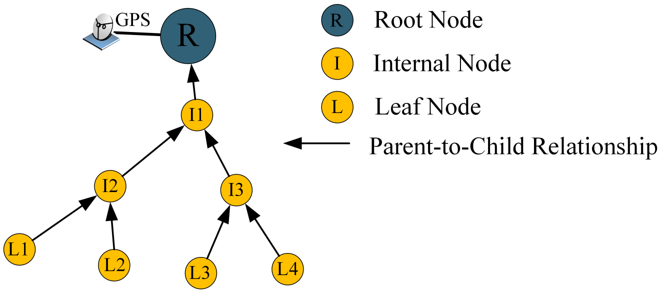

Fig. 1. Cluster-tree wireless sensor network.

III. *Desynchronized* Pulse-Coupled Oscillators Protocol

It is common scenario that the clock signal in the embedded systems is implemented by a *counter* register driven by a crystal oscillator. Assumption is given that the discrete clock is updated periodically at an interval $\tau_0 = 1/f_0$ with the nominal frequency f_0. Let $t[n]$ denote the *reference time* at the n-th clock update event, for a *reference clock*, such as the GPS clock at *root node*, $t[n] = n\tau_0$.

In the free-running mode, the clock state P, increases linearly from zero to threshold defined by variable φ. Once P reaches the φ, P is reset to 0, followed immediately by a *Pulse*. Similarly, in the WSNs, the threshold of a *counter* can be set to the same value of φ to achieve the same time synchronization cycle T. Since the clock state threshold φ is greater than the clock update interval τ_0 (i.e., $\varphi >> \tau_0$), assuming the clock is updated m times during one synchronization cycle yields $\varphi = m\tau_0$ [7]. Furthermore, in the *SYNC* packet, the physical header, MAC header, and MAC payload are encapsulated inside the standard IEEE 802.15.4 frame payload, as shown in Fig. 2. The general format is composed of an IEEE 802.15.4 physical header, MAC header, MAC payload and a check sum (CRC).

Taking PCO periodical resetting behavior into account, at the n-th clock update event, the i-th drifting PCO clock state $P_i[n]$ can be modelled as [7]

$$P_i[n] = t[n] + \frac{\sum_{n=0}^{n-1} \alpha_i[n]\tau_0}{f_0} + \phi_i[n] - \sum_{n=1}^{\lfloor n/m \rfloor} \varphi_i[n] \qquad (1)$$

where floor operator $\lfloor n/m \rfloor$ means the largest integer not greater than (n/m). $\alpha_i[n]$ denotes the frequency deviation of the drifting oscillator at the n-th clock update event. $\phi_i[n]$ is the instant phase noise [7].

In the interactive mode, in addition to the state variable evolves as mentioned above, the clock state is corrected when the node receives the *SYNC* packet from other sensor nodes, and a *SYNC* packet is transmitted immediately if the

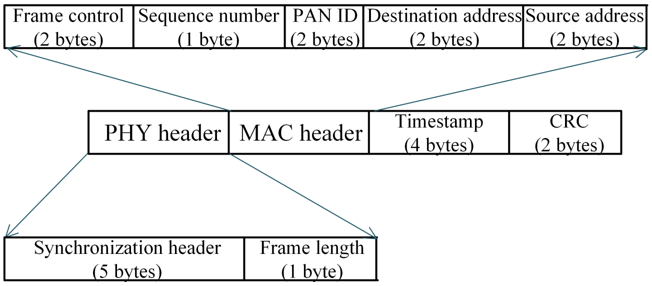

Fig. 2. The format of wireless *SYNC* packet.

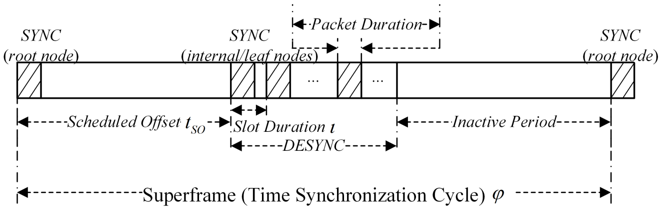

Fig. 3. The Proposed Superframe.

corrected state variable exceeds the threshold. Finally, all the sensor nodes will transmit *SYNC*s simultaneously when synchronization is achieved.

In the large-scale wireless sensor networks consisting of huge amounts of nodes, *SYNC* packets from synchronized sensor nodes (including *internal node* and *leaf node*) interfere with each other and no packet can be received. To avoid this issue, the concept of *desynchronization* is adopted, and all the *SYNC* packets from sensor nodes are therefore transmitted in a uniformly distributed fashion and in the Contention-Free Period (CFP) of IEEE 802.15.4 superframe.

In Fig. 3, the proposed superframe , bounded by neighbouring *SYNC* packets from the *root node*, is made up of three types of periods, namely - Scheduled Offset (SO), *DESYNC* and Inactive Period. The SO is to transmit the data by using the Carrier-Sense Multiple Access with Collision Avoidance (CSMA-CA) mechanism. The *DESYNC* ensures sensor nodes transmit the *SYNC* packets to achieve the DESYNC via the TDMA mechanism. In addition, the Inactive Period is to let the sensor nodes sleep to reduce the power consumption [5].

At the n-th clock update event, by taking implementation of *desynchronization* into account, the i-th drifting PCO clock state $P_i[n]$ of (1) is re-written as

$$P_i[n] = t[n] + \frac{\sum_{n=0}^{n-1} \alpha_i[n]\tau_0}{f_0} + \phi_i[n] - \sum_{n=1}^{\lfloor n/m \rfloor} \varphi_i[n] - t_{d_i} \quad (2)$$

where $t_{d_i} = (t_{SO} + (i-1)\ \iota)$ represents the i-th PCO clock's *desynchronization* offset. Moreover, t_{SO} is the duration of SO, and ι means the *slot duration*. The *packet duration* is the time that a complete *SYNC* packet requires for transmission over the wireless channel.

At the k-th time synchronization cycle, both node i and j fire and transmit the *SYNC* packets at their allocated slot. Upon the reception of *SYNC* from node j, node i reads its local clock, generates a timestamp $\bar{P}_{ij}[k]$ and associates it to the received *SYNC* packet. Once a timestamp is obtained, the offset between two nodes can be determined. Let $\bar{\theta}_{ij}[k]$ denote the offset measurement of node i based on the reception of the j-th node *SYNC*, the offset measurement is

$$\bar{\theta}_{ij}[k] = \begin{cases} \bar{P}_{ij}[k] - \kappa - (t_{d_j} - t_{d_j}) & if \quad \bar{\theta}_{ij}[k] < \frac{\varphi_i[k]}{2} \\ \bar{P}_{ij}[k] - \kappa - \varphi_i[k] - (t_{d_j} - t_{d_i}) & if \quad \bar{\theta}_{ij}[k] \geq \frac{\varphi_i[k]}{2} \end{cases} \quad (3)$$

where κ is the transmission delay.

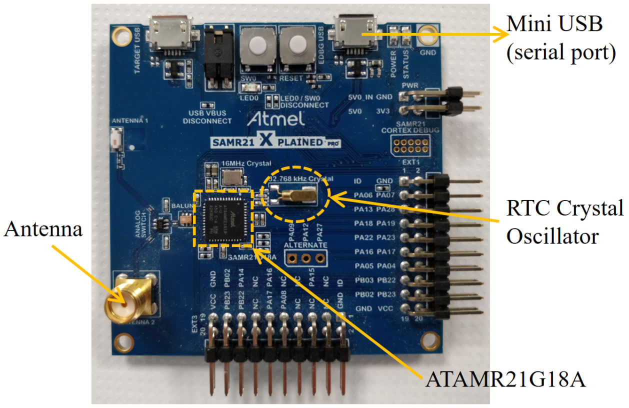

Fig. 4. Prototype of IEEE 802.15.4 radio board.

The determined offset measurement is used to correct the drifting PCO clock directly, and the correction scheme is modelled mathematically by

$$P_i^+[k] = P_i^-[k] - \bar{\theta}_{ij}[k] \quad (4)$$

where $P_i^+[k]$ represents the PCO clock state of an infinitesimal time instant after local clock is corrected at $t[k]$, similarly, $P_i^-[k]$ is the clock state of an infinitesimal time instant before drifting clock is corrected at $t[k]$.

IV. Hardware Platform

The radio board, as detailed in Fig. 4, is employed in this implementation. The Atmel SMART SAM R21 board with low-power 32-bit ARM Cortex-M0+ processor and an integrated ultra-low-power RF233 as 2.4GHz ISM band transceiver. The *SYNC* packet is transmitted between processor and the transceiver via SPI bus (PB31, PB30, PC19 and PC18).

By using the interrupt scheme, the time accuracy of interrupt-based timestamping is not compromised by other tasks (e.g., delays for intensive data processing). At a lower computation burden, the interrupt response is faster and lower jitters and delays can be guaranteed. Therefore, a better time synchronization accuracy can be achieved.

V. Software Architecture at Sensor Nodes

Fig. 5 illustrates the operating mode of a wireless sensor node for PCO time synchronization. In the implementation, the PCO clock state (i.e., *counter* register) of each node is represented by a 32-bit register COUNT of RTC module. When the supply voltage is applied to the sensor node, the wireless node goes into initialization state initializing the RTC and RF modules. Once the procedure of initialization is completed, the sensor node enters the free-running mode, where the COUNT register of RTC module in each sensor node increments linearly toward a threshold value which is defined in the register COMP0. When the COUNT reaches pre-defined threshold value in COMP0, the node will move into the interactive mode, where the COUNT is reset to zero,

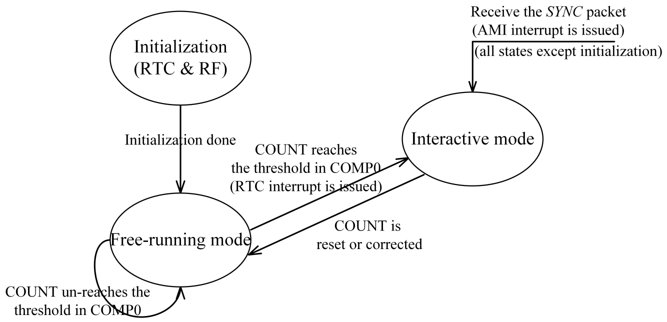

Fig. 5. Operating mode of a wireless sensor node.

and a *SYNC* packet is generated and sent out to the wireless channel. After the COUNT register is reset, the sensor node comes back to the free-running mode to continuously re-count the periodical signal generated by the crystal oscillator from zero to threshold value in COMP0.

There are two ways to generate timestamp, namely, *hardware timestamping* and *software timestamping*. *Hardware timestamping* requires additional hardware, such as DP83640 device from TEXAS INSTUEMENTS and Media Independent interface (MII) between MAC layer and physical layer. It is precise, however is too expensive to be implemented on the low-cost wireless sensor nodes. Instead, *Software timestamping* is widely used in WSNs, where the timestamp is generated by software codes via reading the COUNT register, either in an interrupt or a polling scheme. *Software timestamping* does not require additional hardware, but delays and uncertainties are introduced for the microprocessor to execute the codes of clock reading.

To ensure the good trade-off between timestamp accuracy and hardware cost, *software timestamping* is used by adopting Address Match Interrupt (AMI) scheme and the timestamp is generated in the AMI handler of interactive mode. To be specified, when RF module receives a *SYNC* packet and the AMI is issued, the sensor nodes enter the interactive mode, where the timestamp will be generated by microprocessor via reading the COUNT register of RTC module and filled into the payload format via SPI interface in trigged AMI handler. Once the procedure of filling payload has been completed, wireless packet will be sent to the microcontroller. The 4-byte timestamp format in MAC payload stores the timestamp of wireless sensor nodes receiving *SYNC* packet, as shown in Fig. 2. Moreover, in the AMI handler of interactive mode, once a timestamp is obtained, the offset measurement is determined for clock correction of (4) to achieve the time synchronization in wireless sensor networks. The wireless sensor node will go to the free-running after its COUNT register is corrected.

VI. Results

The *desynchronized* PCO time synchronization is implemented on the WSN testbed shown in Fig. 4. The frequency of the COUNT register on each sensor node is $32.768kHz$ by using the RTC crystal oscillator; to model the Pulse Per Second (PPS) signal, the value of register COMP0 is configured to 32767 and the duration of time synchronization cycle is $1s$. Additionally, the length of *SYNC* packet is set to 21 bytes. The backoff period is configured to zero.

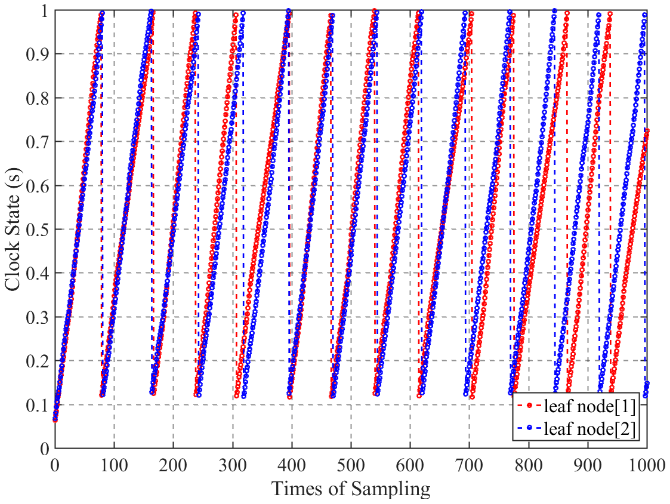

Fig. 6. PCO clock state in the free-running mode.

Fig. 6 indicates the clock state of two sensor nodes in free-running mode, the clock state increase from zero to threshold value of 1 second (i.e., the register COMP0 of 32767), and it is reset after the threshold value is reached. Therefore, the *desynchronized* pulse-coupled oscillators are implemented on the hardware testbed successfully.

In Fig. 7, the *root node* broadcasts the *SYNC* packet each second, after the broadcasted *SYNC* is received by four *leaf node*s, the AMI interrupt is issued to generate the local timestamp via reading the COUNT register of RF module, the experimental results coincide with the simulation results of [6] that as the time increases, the clock offset increases

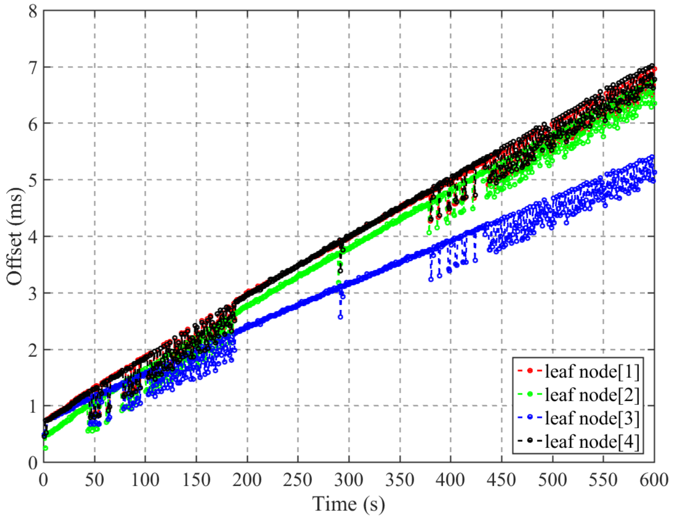

Fig. 7. Clock offset in PCO free-running mode.

to a value which depends on the clock operation time and the intrinsic stability of the unregulated clock. Therefore, the synchronization scheme is needed to correct the drifting clock to achieve the time synchronization in the wireless sensor networks.

Conclusion

In this paper, the *desynchronized* PCO pulse-coupled oscillators model is implemented on the hardware testbed successfully, which is compatible with existing IEEE 802.15.4, to enable the PCO extension in the large-scale wireless sensor networks. As the basis of future research work, the performance of both *desynchronized* pulse-coupled oscillators and classical pulse-coupled oscillators will be evaluated in a real-world context.

Acknowledgment

The authors acknowledge Yuchen JI for valuable technical support.

References

[1] D. Buranapanichkit, N. Deligiannis, Y. Andreopoulos., “Convergence of Desynchronization Primitives in Wireless Sensor Networks: A Stochastic Modeling Approach,” IEEE Transactions on Signal Processing, vol. 63, issue 1, pp. 211-233, Jan 2015.

[2] J. Degesys, I. Rose, A. Patel, R. Nagpal., “DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks,” in International Conference on Self-Adaptive and Self-Organizing Systems, Cambridge, USA, 2007, pp.11-20.

[3] R. Gentz, A. Scaglione, L. Ferrari, and Y. Hong., “PulseSS: A Pulse-Coupled Synchronization and Scheduling Protocol for Clustered Wireless Sensor Networks,” IEEE Internet of Things Journal, vol. 3, issue 6, pp. 1222-1234, Dec 2016.

[4] R. E. Mirollo and S. H. Strogatz., “Synchronization of Pulse-Coupled Biological Oscillators,” SIAM Journal on Applied Mathematics, vol. 50, issue 6, pp. 1645-1662, 1990.

[5] Y. Zong, X. Dai, Z. Gao., “A Software Simulator of Discrete Pulse-Coupled Oscillators (PCO) Time Synchronization in Wireless Sensor Networks,” in International Conference on Automation and Computing (ICAC), Huddersfield, UK, 2017, pp.1-7.

[6] Y. Zong, X. Dai, Z. Gao, R. Binns, K. Busawon., “Simulation and Evaluation of Pulse-Coupled Oscillators in Wireless Sensor Networks,” Systems Science and Control Engineering, vol. 6, issue 1, pp. 337-349, Jul 2018.

[7] Y. Zong, X. Dai, Z. Gao, K. Busawon., “Modelling and Synchronization of Pulse-Coupled Non-identical Oscillators for Wireless Sensor Networks,” in International Conference of Industrial Informatics (INDIN), Porto, Portugal, 2018 (to be published).