VOL. 12, NO. 9, MAY 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

TIME SYNCHRONIZATION FOR WIRELESS NETWORKS USING ZigBee AND Zynq FPGAs

M.Vishwanath[1], Himani Goyal[2] and Habibulla Khan[1]
[1]Department of Electronics and Communication Engineering, KL University Vaddeswaram, Guntur, India
[2]Regional College for Education Research and Technology, Jaipur, Rajastan, India
E-Mail: mvisu87@gmail.com

ABSTRACT

The last decade has witnessed immense increase of wireless networks for different applications in mobile communication, internet, and industrial automation. The next generation wireless nodes will be used in real time applications such as auto navigation, cloud computing and ad-hoc data networks etc. These applications need precise time synchronization to function properly. In this paper, the time synchronization between two wireless nodes is achieved using novel technique. The main principle is to periodically synchronize clocks running in two different nodes through wireless communication protocols. Each node consists of one ZigBee module working at 2.4 GHz ISM (Industrial, Scientific and Medical) band radio and Zynq Configurable System on Chip (C-SOC) platform. Based on the ZigBee communication the Field Programmable Gate Array (FPGA) running real time clock with resolution of 10 nano seconds is synchronized through software. The modules developed in this experiment are FPGA real time counter, External Trigger interface module and UART communication. To validate the developed method another Spartan 3E FPGA generated trigger pulse is used. The time synchronization error is among the two wireless nodes is analyzed and plotted using Matlab. The results demonstrated reduction of time synchronization error up to 10 micro seconds. The work is aimed to be extended towards realizing other techniques for further improvement in the time synchronization.

Keywords: time synchronization, wireless networks, FPGA, Zynq, C-SOC, UART, ZigBee.

1. INTRODUCTION

The current generation wireless applications such as Auto Navigation, Cloud computing, Ad-hoc wireless networks demand precise time synchronization. The accuracy of time synchronization will determine whether the wireless network is able perform well. Each node in wireless network need to know at what time instants the events is happening, in synchronization with all other nodes. Time synchronization plays an important role in wireless communication networks. When designing time synchronization algorithms for wireless communication networks, a number of parameters need to be carefully considered, including precision, processing time overheads and energy costs. The objective of the proposed work is to investigate the precise time synchronization algorithms for wireless communication systems. The work also aims to investigate best possible software based time synchronization mechanism for wireless networks. As the Zynq 7000 series FPFA is best suitable platform, the realization aspects on Xilinx Zynq 7000 FPGA will be investigated. The application level time synchronization schemes will be studied on Zynq C-SOC based platform and best achievable time synchronization on Zynq will be attempted. The below section discusses few similar research works.

The work given at [1] discussed about micro second level time synchronization in Wireless sensor networks and focuses on known synchronization protocols like TPSN, FTSP, RBD, GTSP etc which ignores pocket handling jitters, clocking drifting and pocket loss. Works concludes with “off-set” only implementations do not permit long time synchronization. The research work [2] does not correct the drifts and skew of the clocks of the nodes. Instead uses the fact that the time interval between two transmitted beacons is same as the time interval between two receiver beacons, because of propagation and coding delay are assumed to be same. Work is done using Medium Access Control (MAC) protocol. Paper [3] suggests that external hardware can be used to synchronization of the clocks and reduce the clock drifts without intervention of the host Microcontroller (MCU), the actual time sync circuit continuously trim the MCU low power clock and periodic discrete updates from MCU can introduce interpolation errors.

The other work [4] investigates a system called RT-WIFI which is TDMA data link layer protocol based on IEEE 802.11 physical layer to provide high sampling rate and deterministic timing guarantee on pocket in wireless control systems. Time synchronization may be done for hybrid networks (Wired and Wireless) targeting real time communication systems in factory automation. Similar work is presented [5] to experiment with IEEE 802.11 hybrid network with IEEE 1588 synchronization protocol and employing software time stamping.

Proposed method at [6] is low-cost and improved precision time stamping technique based on Ultra Wide Band (UWB) signaling. Work claims high accuracy of time synchronization by getting advantage of UWB signaling immunity to interferers and multipath. Work presented [7] to have customized hardware design to consider factors like precision and processing time overhead. Work focuses on various other aspects like Hardware interfacing low-level event time stamping with inclusion of software implementation. Other works [8] presents the idea of using existing sync protocols and improvising to get more accuracy and control in time synchronization. Work focused on IEEE 1588 standardization. One of the important aspects of any time

synchronization system is handling and managing the time stamping. Paper [9] works on adding time stamp in physical layer. Work concludes by claiming the reduction of influence of non determinacy factor to precision synchronization.

The remaining part of the paper is organized into 5 sections. Section 2 presents about high level architecture of proposed technique, section 3 has implementation details. Section 4 has simulation and hardware verification results. Final section concludes the work.

2. HIGH LEVEL ARCHITECTURE

The Figure-1 shows high level architecture of time synchronized wireless node with the proposed scheme.

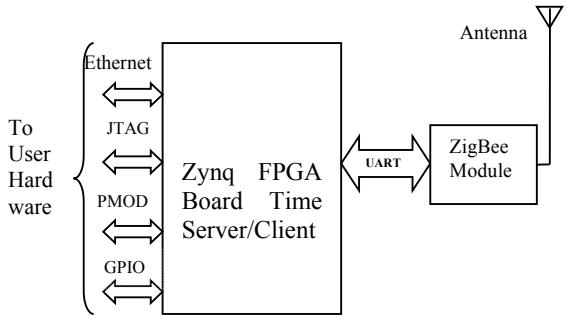

Figure-1. Block diagram representation server/Client time synchronization module.

The developed Time synchronization wireless node is designed to cater for multiple categories of applications. The Zynq FPGA board which acts like Time sync Server/Client is capable of handling user hardware connectivity by multiple protocols. The 1 Giga bit Ethernet, JTAG, General Purpose Input Outputs (GPIO's) UART and PMOD connectors are available for the user hardware connectivity. The Time Server/Client is realized through Zynq FPGA based solution (ZYBO Board XCT7010T-1CLG400). The ZYBO Board consists the following major ICs and peripherals.

- Zynq FPGA XCT7010T-1CLG400
- One Giga bit Ethernet
- 512 MB DDR3 Memory
- Dual role HDMI interface
- USB OTG 2.0 etc.

The ZigBee wireless module connects to the Zynq FPGA Board through UART protocol.ZigBeewireless module is realized through XB24-8WI-001. This module uses 2.4 GHz antenna for communication.In a multi node wireless networkeach nodeconsists one time synchronization module which is illustrated above. One of these nodes will be configured as a time server. All the remaining nodesact like time synchronization clients. The time server continuously sends time synchronization wireless messages to all the clients. The time synchronization packet consists of local time data. All the clients upon the reception of wireless packet shall synchronize their local clocks with respect to the received time. This mechanism can be repeated at regular intervals to ensure the periodic time synchronization among the total network. Depending upon the accuracy required for the given application this periodicity can be decided. In this research work, it clearly shows by making this time synchronization at regular intervals the overall accuracy can be improvised.

To effectively test and verify the propose scheme a setup has been established, which is illustrated in Figure-2.

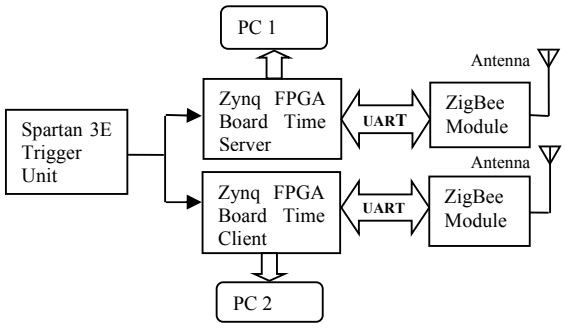

Figure-2. Block diagram representation of proposed test setup.

To measure the time synchronized error it is proposed to send a reference pulse to two wireless units. Upon the reception of trigger pulse each node shall log their local timestamp. Each node is connected to Personal Computer (PC) through UART to log the timestamp information. The logged data in both the PCs is brought into one computer and plotted in Matlab for measuring the error for each triggerinstance. Details are given in further sections.

3. IMPLEMENTATION DETAILS

The below figure shows Registered Transfer Level (RTL) diagram of implemented VHDL module, which is common for both server and client nodes. This setup implements Timing module. In the initial level the timing module is developed without the synchronization capability. This module developed using VHDL language and synthesized for Zynq FPGA.

A. Timing module without synchonization

The Zybo FPGA board provides 100 MHz clock which is used to run the total design. The input 100 MHz clock is applied to a divide by 100 counter to produce a 1 MHz clock. The rising edge detector generates a 10 nano second period for every 1 micro second period. This pulse is used to run the 48 bit counter. To measure the error between two synchronization nodes additional logic is added which is explained below. As shown in Figure.2 the Spartan3 GPIO output is taken on two wires and both are fed to two different nodes.

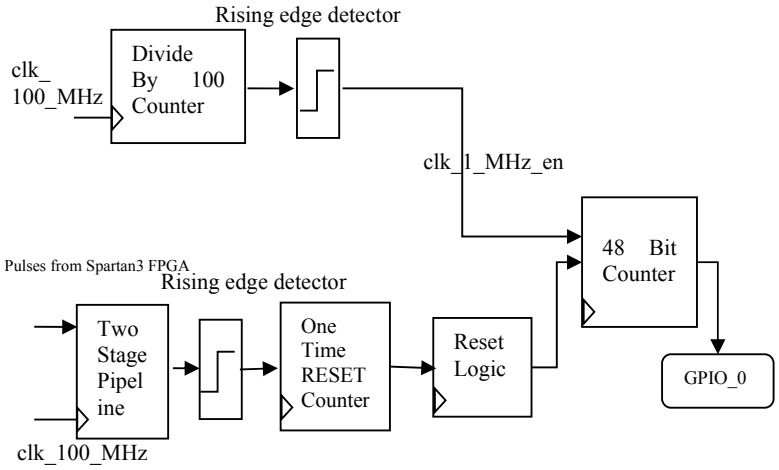

Figure-3. Timing module without synchronizing capability.

The Spartan3 generates 3.3 Volts pulse for every 10 seconds interval. Figure-3 shows the second data path, which utilizes the Spartan3 generated pulse. To avoid false triggers Spartan3 generates a three pulse pattern. The two stage pipeline followed by rising edge detector block detects the three pulse pattern and generates pulse to next stage block. The one time reset counter and reset logic generates a pulse to 48 bit counter. The ARM 9 Processor of Zynq FPGA continuously polls for data availability from RTL and reads whenever the 48 bit counter value made available. The read value by the processor is sent over Universal Asynchronous Receiver Transmitter (UART) to PC for further analysis.

B. Timing module for Server (with synchronization)

The Figure-4 shows timing module on Server side with time synchronization capability.

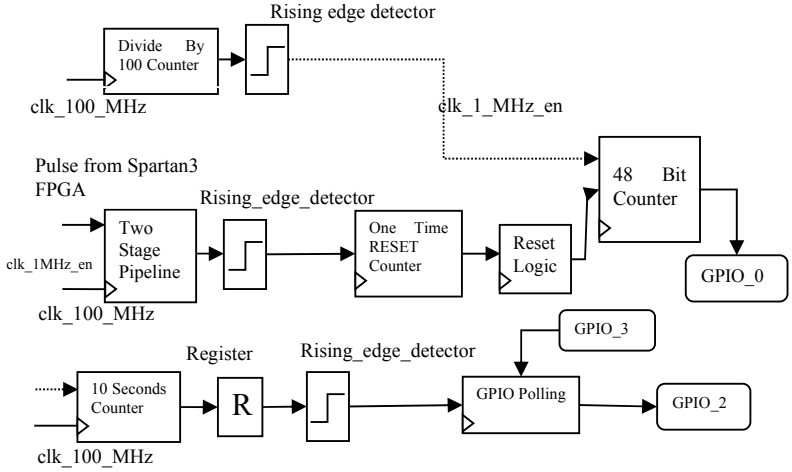

Figure-4. Timing module (server) with synchronization.

The third data path which is added in timing module provides the synchronization capability. The 1 MHz counter pulse is used to count 10 seconds. Upon the expiry of ten seconds count the RTL logic shows its readiness for sending time synchronization data. The processor in time synchronization node continuously polls for data from RTL. This is indicated through by setting a 1 in bit position 4 of GPIO_3. Once the ten seconds counter is expired RTL logic shows its readiness by setting one in GPIO_2 bit position 1. The processor running software module continuously checks for bit position 1 of GPIO_2. As on when this bit become one immediately it reads the GPIO_0 which consists of server side local time and transmits over ZigBee. The Figure-5 shows software implanted on server node.

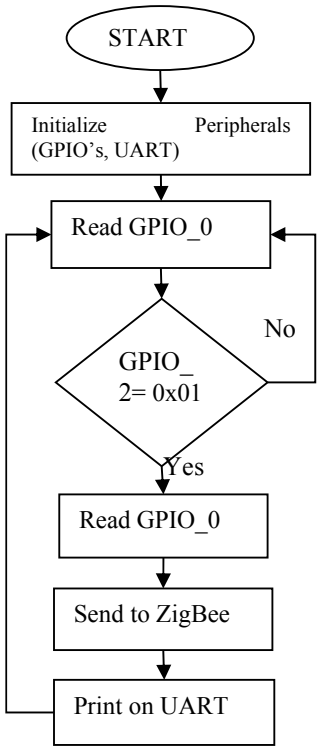

Figure-5. Flowchart server side software.

Software execution flow begins with initializing peripherals called GPIOand UART. Now data from GPIO_2 will be read to verify its Least significant bit data that is whether it is '0' or '1'. If it equals to '1' software will read the time value from GPIO_0, otherwise it will wait until GPIO_2 flag goes '1'. Once the data is read from GPIO_0 it is sent to ZigBee unit to transmit it to Client node, at the same time same data is sent to UART within Server unit.

C. Timing module for client (with synchronization)

The following Figure shows timing module for client side. The first data path is same as that of server module to generate a 1 MHz pulse. The second data path provides a capability of loading externally received time sync data. Upon the reception of ZigBee message the client side software sets GPIO_3 bit position 0.

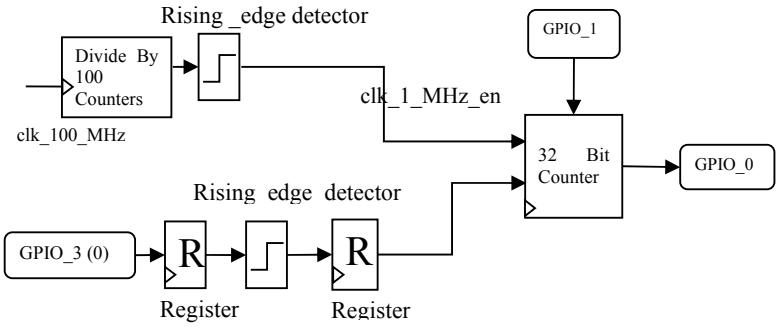

Figure-6. Timing module (client) with synchronization.

The RTL logic detects 1 on GPIO_3 and loads the external time sync data available from GPIO_1. Figure-7 shows flow chart for ARM 9 implemented software on client side.

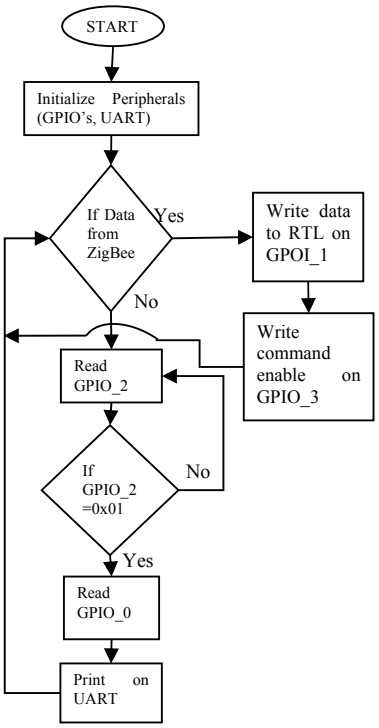

Figure-7. Flowchart for client side software.

Similar to server node it begins with initializing peripherals like UART and GPIOs. Once the initialization phase is complete it checks for the data received from ZigBee. If yes, the same will be loaded into GPIO_1 which is intern update the RTL 32-bit timer register. At the same time for handshaking a command enable signal is made available into GPIO_3. Later it will be taken as enable signal in RTL of Client node timing module. In case of no data received from Server through ZigBee module data from GPIO_2 will read. This state of process continuously polls for GPIO_2 until it is equal to '1'. Once the GPIO_2 least significant bit is equal to '1' then data from GPIO_0 will be read which is time value of Client node timing module and same will be written to UART for sending the data to PC and flow continue with continuously checking for updated ZigBee data received in Client node from Server node.

4. SIMULATION AND HARDWARE VERIFICATION

This section shows the VHDL level simulation and verification of developed RTL modules. The figure shows simulation of server side timing module.

A. Simulation results

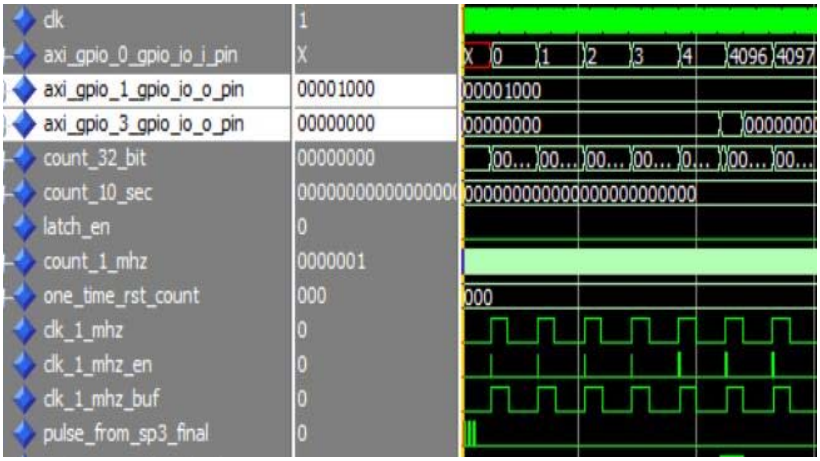

Figure-8. Simulation results of server side timing module.

As illustrated in Figure-8 it shows that as the request from client arrives on GPIO_3, Server software checks for the request at GPIO_3 of bit position 0. After detecting rising edge detection in RTL (as mentioned in Figure-4) 48 bit counter value is compared and updates the client counter value with the drift free time value. This can be observed in Figure-7.

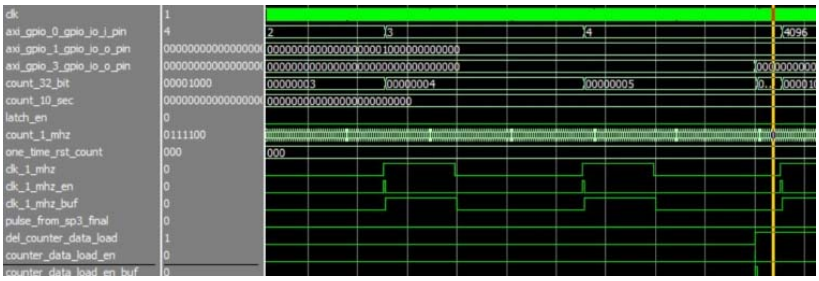

Figure-9. Simulation results of server side timing module of updated time value.

B. Hardware verification

The below figure shows the Hardware setup to verify the proposed time synchronization technique. The hardware shown on the left side is Numato Board with Spartan 3E FPGA for generating timing pulse. The two Zybo board are connected with ZigBee modules. The wiring from this board to two wireless time synchronization nodes can be observed. The two time synchronization nodes (which are present in right side in below Figure) have been seen. The two ZigBee modules can be observed.

Figure-10. Hardware verification setup using client server nodes along with Spartan3 module.

The below Figure shows the setup of two PCs used capture time data from two nodes. On each PC Xilinx SDK software is used to capture the data. The UART baud rate settings are 9600. The timing pulses, which are generated by Spartan3 FPGA, are of 3.3 Volts TTL pulse which is used to give trigger to both the Zybo boards.

Figure-11. Hardware verification setup using client server nodes with ZigBee modules with synchronization.

High speed cabling is provided bewteen the boards to ensure proper transmision of the pulses.

C. Verification and analysis

Both side PC received UART messages are taken one Computer and using Matlab further analysis is carried out. In Matlab both nodes generated file data is read and error between them is computed. The error is plotted as function of time. The following figure shows the error plot for no synchronization case.

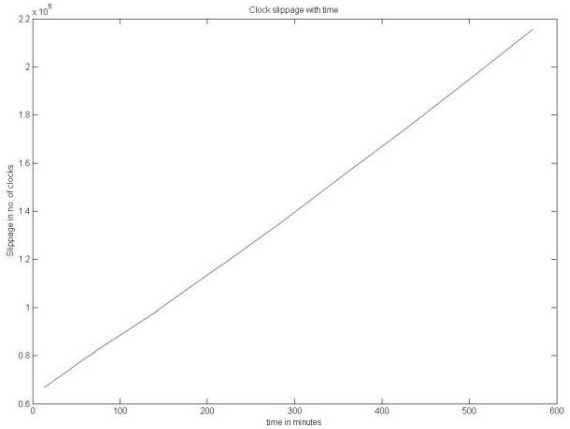

Figure-12. Error plot for two wireless nodes without time synchronization.

After establishing synchronization capability the experiment is repeated and error analysis is carried out. The Figure below shows the error between two nodes with synchronization capability.

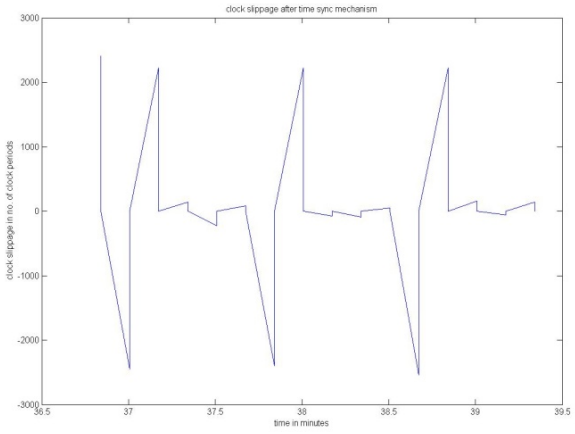

Figure-13. Error plot for two wireless nodes with time synchronization.

It is apparent from the Matlab Analysis that with synchronization the peak error is always limited to +/- 20 µ seconds over the entire period of testing. Whereas for no synchronization case the time mismatch between the two nodes continuously grows. It is observed that over the interval of time of 10 hours the peak error without synchronization is led up to 200 micro seconds.

Table-1. Time synchronization error table.

Test case	Time synchronization error	
	Test condition	Peak error
1	Without synchronization	200 µseconds
2	With synchorinization	+/- 20 µ seconds

5. CONCLUSIONS

The present paper discusses the necessity of hard real time synchronization between the wireless nodes. The setup is developed to illustrate the timing mismatch between two nodes when there is no synchronization between them. The paper presents a time synchronization mechanism with low cost ZigBee wireless modules. The present setup utilizes latest Zynq System on Chip (SoC) architecture consisting both processor and FPGA on single chip. The results show improvement in error limited only to +/- 20 µ seconds with time synchronization over long periods of time. Whereas the errors with synchronization were observed continuously growing even more than hundreds of micro seconds. Work is aimed to continue in similar direction to further investigate synchronization capabilities and associated challenges and robust synchronization mechanism between wireless sensor nodes.

REFERENCES

[1] Djamel Djenouri, Miloud Begaa. 2014. Implementation of High Precision Synchronization Protocols in Wireless Sensor Networks. WOCC2014. IEEE Transactions.

[2] Tao Ma, Zhanqi Xu, Michael Hempel, Dongming Peng, and Hamid Sharif. 2014. Performance Analysis of a Novel Low-Complexity High Precision Timing Synchronization Method for Wireless Sensor Networks. IEEE Transactions on Wireless Communications. 13(9).

[3] Maxim Buevich, Niranjini Rajagopal and Anthony Rowe. 2013. Hardware Assisted Clock Synchronization for Real Time Sensor Networks. IEEE 34th Real-Time Systems Symposium. IEEE Computer Society.

[4] Yi-Hung, Quan Leng, Aloysius K. Mok, Wenlong Zhang, and Masayoshi Tomizuka. 2013. RT_WiFi: Real-Tie High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications. IEEE 34th Real-Time Systems Symposium. IEEE Computer Society.

[5] Aneeq Mahmood, Felix Ring. 2012. Clock Synchronization for IEEE 802.11 based Wired-Wireless Hybrid Networks using PTP. IEEE Transactions.

[6] C.M De Dominicis, A. Flammini, S Rinaldi, E.Sisinni, A. Cazzorla, A Moschitta, and P Carbone. 2011. High Precision UWB-based timestamping. IEEE Transl.

[7] Nason Tackett, Emil Jovanov, and Aleksandar Milenkovic. 2011. An Implementation of Time Synchronization in Low-Power Wireless Sensor Networks. IEEE Transl. 971-1-4244-9593-1//11.

[8] Darold Wobschall and Yuan Ma. 2010. Synchronization of Wireless Sensor Networks Using a Modified IEEE 1588 Protocol. IEEE Transl. 978-1-4244-5977-3/10.

[9] Ping Song, Xiaodong Shan, Kejie Li and Guangping Qi. 2009. Multi-hop Based Highly Precise Time Synchronization Protocol for ZigBee Networks. IEEE Transl. 978-1-4244-4076-4/09.