AN EXPERIMENTAL STUDY ON TIME SYNCHRONIZATION IN LINEAR SPANNING TREE WIRELESS SENSOR NETWORK

A MASTER'S THESIS

in

Electrical and Electronics Engineering

Atılım University

by

AHMET ERPAY

SEPTEMBER 2018

AN EXPERIMENTAL STUDY ON TIME SYCHRONIZATION

IN LINEAR SPANNING TREE WIRELESS SENSOR NETWORK

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

AHMET ERPAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2018

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

____

Prof. Dr. Ali KARA

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Efe ESELLER

Head of Department

This is to certify that we have read the thesis Time Synchronization in Linear Spanning Tree Wireless Sensor Network submitted by Ahmet Erpay and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Prof. Dr. Ali KARA

Supervisor

Examining Committee Members

Prof. Dr. Bülent TAVLI ______________________

Prof. Dr. Ali KARA ______________________

Assoc. Prof. Dr. Enver ÇAVUŞ ______________________

Assoc. Prof. Dr. Mehmet ÜNLÜ ______________________

Asst. Prof. Dr. Mehmet Efe ÖZBEK ______________________

Date: 26.09.2018

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ahmet ERPAY

Signature:

ABSTRACT

TIME SYCHRONIZATION IN LINEAR SPANNING TREE

WIRELESS SENSOR NETWORK

Erpay, Ahmet

M.S., Electrical and Electronics Engineering Department

Supervisor: Prof.Dr. Ali Kara

September 2018, 32 pages

In this study, time synchronization in a linear spanning tree wireless sensor network is studied. The method is applied on a PIC based platform where time synchronization process and temperature effects on it are focused. Outdoor experiments are fused with temperature change and the aim is to see performance of the wireless sensor network and to overcome possible challenges.

Keywords: Time synchronization, Clock drift, Wireless sensor network, Spanning tree network

ÖZ

ÇİZGİSEL YAYILAN AĞAÇ KABLOSUZ ALGILAYICI AĞDA EŞZAMALAMA

Erpay, Ahmet

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ali Kara

Eylül 2018, 32 sayfa

Bu çalışmada, çizgisel yayılan kablosuz algılayıcı ağda eşzamanlama çalışılmaktadır. PIC tabanlı bir platform üzerine metot uygulanırken eşzamanlama ve eşzamanlama üzerindeki sıcaklık etkileri odak alınmıştır. Açıkhava deneyleri sıcaklık değişimleri ile birleştirilip, olası engelleri bertaraf etmek için kablosuz algılayıcı ağın performansını takip hedef olarak seçilmiştir.

Anahtar Kelimeler: Eşzamanlama, Saat kayması, Kablosuz algılayıcı ağ, Yayılan ağaç ağı

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARQ	-	Automatic Repeat Request
ATS	-	Average Time Synchronization
DMTS	-	Delay Measurement Time Synchronization
FEC	-	Forward Error Correction
FTSP	-	Flooding Time Synchronization Protocol
LTS	-	Lightweight Tree-based Synchronization
MAC	-	Medium Access Control
MCU	-	Microcontroller Unit
PIC	-	Peripheral Interface Controller
PPM	-	Parts Per Million
RBS	-	Reference Broadcast Synchronization
RF	-	Radio Frequency
SNTP	-	Simple Network Time Protocol
TDP	-	Time-Diffusion Synchronization Protocol
TI	-	Texas Instruments
TPSN	-	Timing-Sync Protocol for Sensor Network
USB	-	Universal Serial Bus
WSN	-	Wireless Sensor Network

1. INTRODUCTION

Wireless sensor networks (WSNs), which are distributed networks consisting of low-power devices capable of monitoring, sensing, data processing and communication, gather increasing attention from both academia and industry in the past decade [1]. WSNs can be used in various areas like industry, military, medical and environmental domains. WSN provides security and this is one of the major reasons for its wide use. Considering all these domains every sensor device, which is also called sensor node, should sense and process data according to a common time to interpret information meaningfully. Time synchronization is used to create common timescale in network. Synchronized networks can transmit data in scheduled manner and use sleep routine for power efficient systems. For example, an environmental monitoring application's sensed data should be fused with right timestamps, otherwise the fused data will be unusable. It is also important to use their power efficiently for this purpose all nodes have to sleep and wake up all together where time synchronization is a must.

Nodes' clock model is assumed as a linear function of time, hence clock rate is also assumed constant [2]. This model is reasonable for a period of time and synchronization algorithms can be built upon this. On the other hand, clock rates may drift from each other with time and offset occurs between them. Combining these two point of views, in a WSN time synchronization is periodically done to maintain common timescale.

To estimate offset and drift changes between nodes there are many studies on literature. Nodes can synchronize between them by single-hop or multi-hop networks. If a node directly synchronize with another node in the network this structure is called single-hop network. On the other side, if two nodes can not reach each other directly and need one or more node to synchronize this is called multi-hop network. Methods can be divided according to these network structures. Reference Broadcast Synchronization (RBS) is a single-hop type method where master node broadcasts beacons and other benefits from that periodic beacon, is proposed in [3]. While RBS is freed of some sender-based uncertainty, its scalability is not big. Flooding Time Synchronization Protocol (FTSP), which is also single-hop type, is presented in [4]. FTSP method can choose a root node dynamically and this node broadcasts periodic messages to network. Every node that takes this message rebroadcasts it while takes a time stamp at every received message. FTSP has accommodation in network topology variations and robust against failure of nodes. It is also energy efficient. Delay Measurement Time Synchronization (DMTS) is single-hop type where leader node broadcasts its time, many other nodes take this message and measures the delay and adds it to leader's time to set their own time [5]. It is energy efficient and simple but precision is not high. All single-hop type methods can work in multi-hop type with some changes in their methods. The following methods are all multi-hop types: In [6] Timing-Sync Protocol for Sensor Networks (TPSN) is presented. This method has two phases which are level discovery and synchronization. In level discovery phase, one node is selected as a root (level 0) and others are assigned to different levels due to closeness of the nodes. In synchronization phase, nodes synchronize with pair-wise synchronization method from level 0 to last level. TPSN has good accuracy but does not allow dynamic topology. Lightweight Tree-based Synchronization (LTS) which is presented by Greunen and Rabaey[7] is constructed on pair-wise synchronization. LTS has two different approaches: First one starts with creating a spanning tree with a sink node that has some reference point. Then it uses n-1 pair-wise synchronizations for n nodes in the network. It is flexible method but complexity can be high or low due to sink node. In [8] Average Time Synchronization (ATS) is presented where average times are found on every node in network by using pair-wise message exchange. In ATS the sequence of averaging nodes is important and uses more energy to classical pair-wise process. In Time-Diffusion Synchronization Protocol (TDP) there is an equilibrium-time that is agreed on network by all nodes and all local clocks are bounded around this time with a small deviation [9]. TDP is flex and fault tolerant but needs high time for convergence.

The given methods, which have many experimental results on both single-hop and multi-hop [10-13], are generally indoor placed and run in ideal conditions. It is seen that in many cases temperature, RF interference and true conditions are neglected at implementing methods on different motes. On the multi-hop side, while there are different topologies are established experimentally, there is no line topology structure is tried. Experiment places like rural or silvan areas are not used for testing where sensor nodes are used in such places.

The aim of this study is establishing time synchronization along a spanning tree WSN. A natural gas pipeline is used to transfer natural gas from origin to demanding areas quickly and efficiently. To ensure that, pipeline has to work properly, pipes should be protected. Pipes are severely damaged from corrosion over the years. There are some methods to counter this problem. These methods are anodic protection, cathodic protection, corrosion inhibitor and coating. Cathodic protection is the most common method, that is applied on corrosion problems. In cathodic protection method, the idea is to prevent the accumulation of Oxygen on the metal structure where the metal is chosen as cathode and accumulation is done on the anode. Cathodic protection measurements need time based measuring devices. In cathodic protection systems, anode and cathode (pipe) structures that are placed on consecutive measurement points along pipeline, need removing of conductors between them during measurement, because there is a conductor between anode and cathode parts for cathodic protection. It is important to remove these conductors on every measurement point at the same time interval to maintain a homogeneous depolarization along pipeline. In a measurement type which is called "Instant-off", electrical saturation of the pipeline is measured and it is done by cutting out anodes of depolarization's all measurement points at the same time interval. This time interval can be as long as 100 millisecond for healthy measurement. All the measurements have to be done in this time interval. These measuring points are placed on a pipeline with various distances that may vary between 300 m and 2000 m. It is essential to synchronize these devices' time on a common time scale to provide measuring process in the same time interval. These devices' positions on the pipeline create a line and a device can communicate well with previous and next device on a WSN. This geographical arrangement of the devices directs system to use hierarchical line structure where it is seen that a spanning tree WSN is best choice to build a network. Geographical effects and temperature variations on time synchronization are observed and a distributed lightweight time synchronization algorithm is used.

In Section 2, Wireless Sensor Networks (WSNs) are defined and their structure is introduced. WSNs' topological behavior is mentioned.

In Section 3, time synchronization methodology is clearly explained in the light of topological approach. Crystal oscillators' effect on clocks and nodes are referred. Experimental implementation and procedures are also presented.

In Section 4, results are given. Their inferences and expectations are interpreted.

In Section 5, results are analyzed and evaluated. Lastly, conclusion of the study is given.

2. WIRELESS SENSOR NETWORKS

Wireless Sensor Networks (WSN) are formed of small low power and low cost devices that can sense their surroundings such as temperature, humidity, sound and light[14]. WSN devices are called as nodes (sensor nodes) and these nodes can form networks without any infrastructure. As these nodes are cheap and tiny, they can be placed at many different areas to monitor different type of situations.

A WSN has many nodes in it and generally, at least one of them is used as base station to collect processed data from other nodes. The interaction of nodes can change according to network structure and the transfer of data can be done in single-hop or multi-hop. If one or more of the nodes die, a multi-hop network has more chance than the single-hop network on maintaining its vital functions.

WSNs' improvement is directly related to its parts and they are evolving to next stages as technology advances. As Industry 4.0 emerges, WSNs can have place in smart factories and intelligent manufacturing systems [15].

Below, hardware and topological structure of WSNs are explained.

2.1 Hardware Structure

A sensor node is a device that collects data from various sensors and performs some processing on these data. It communicates with other nodes according to these processed data and the communication between nodes can change due to network requirements. A basic node is generally formed of four main parts as seen in Figure 2.1.

A basic node consists battery, microcontroller, sensors and transceiver. These parts can change or fuse in a single unit in need. For example, MCU can have internal memory or a memory unit can be added to this basic structure. Sensors can be basic sensors that give analog output or they have an own MCU where some pre-processing is done. Battery may have some energy harvesting parts or it is simply single use only. These changes are all related to the needs of WSN.

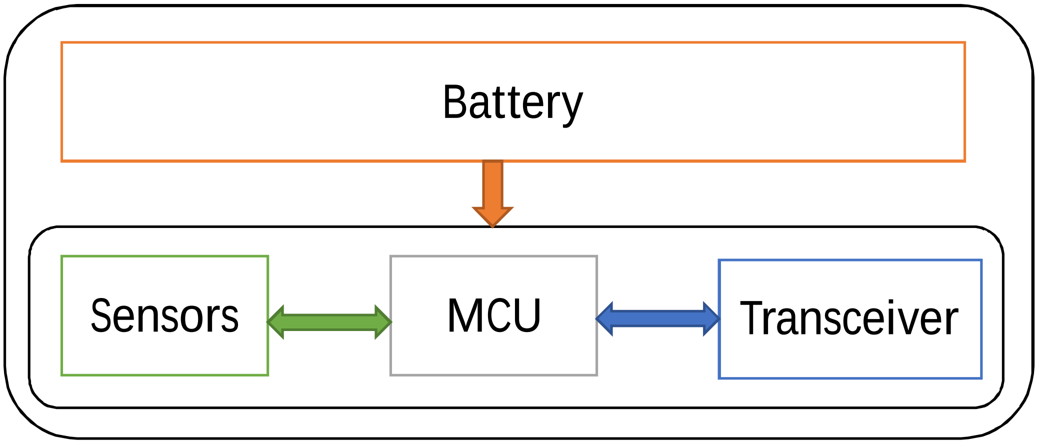

Figure 2. 1. Architecture of the sensor node.

The biggest challenge of a sensor node is energy and to overcome this problem node should be in sleep mode whenever its job is done. Communication unit in a sensor node is the most energy spending part. The transceiver is turned on whenever a transmit or receive process is need. The protocol stack is used to control these processes where it has five layers:

Physical Layer: It is responsible for frequency settings, signal detection, modulation and encryption.

Data Link Layer: It is responsible for data streams, frames and medium access. In WSN medium access control (MAC) protocol is widely used to handle multi-hopping and self-organization. MAC helps WSN communication and shares communication resources in network. On the other hand, MAC's power consumption is the biggest disadvantage and several MAC protocols have been developed to prevent this [14]. Error control is also one the important part of data link layer. Forward error correction (FEC) and automatic repeat request (ARQ) are vital parts of error correction. Complexity of FEC is a problem for a sensor node so basic encoding and decoding options are more appropriate.

Network Layer: It is responsible to route data from one node to another node or sink node via other nodes or directly. Power efficiency is vital for choosing right route on networking. For connecting to another network, a backbone is also created in this layer.

Transparent Layer: If network will connect to internet or another network this layer is needed. Since sensor nodes have different constraints on hardware, development of this layer is not easy and can change according to different WSNs, which have different types of nodes.

Application Layer: In this layer main application and other management functions are included.

This layered architecture[14] has many advantages but it also comes with non-deterministic delays between layers make it hard for time synchronization. On some WSNs bare metal programming is used to minimize these delays and to control whole life cycle of a sensor node.

2.2 Topological Structure

In the protocol stack, network layer is mentioned. It is important to understand its structure since a WSN is built on this. A network is a collection of nodes, computers, servers, peripherals or other devices where they are linked to each other in order to share data. These links are established over wires, optic cables or wireless media. Network topology is the physical or logical schematic description of network arrangement. It states how the connections are done between nodes. Network topology is divided into physical topology and logical topology. Physical topology concerns about physical layout of the connected nodes, while logical topology focuses on how data flows in the network. Different logical topologies can be built in the same physical topology.

Physical topology can be categorized into seven models: Point-to-point, bus, ring, star, mesh, tree and hybrid:

Point-to-point topology is simplest and easiest network. This network can be used in other topologies or different networks can be linked to each other with this structure.

Figure 2. 2. Point-to-point network topology

Bus topology is a network type where every node in the network is connected to a single cable. It has terminators on each end of the bus and cable length has a limit. Bus topology is simple and cheap, but traffic is heavy on common line and network fails if backbone cable fails. This network is slower than ring network.

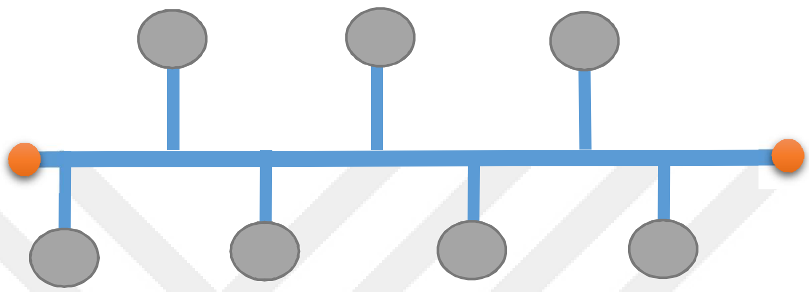

Figure 2. 3. Bus network topology

Ring topology is formed of nodes that connect each other like a ring. Data flow is unidirectional and it can be bidirectional by having two connections between each nodes, where this is called dual ring topology. Data has to pass each node till reaching destination node by this data signal is repeated at every node and remains strong. Ring topology can handle high traffic and is cheap to install. On the other hand, adding, removing or failure of a node disturbs whole network and to troubleshoot is not easy.

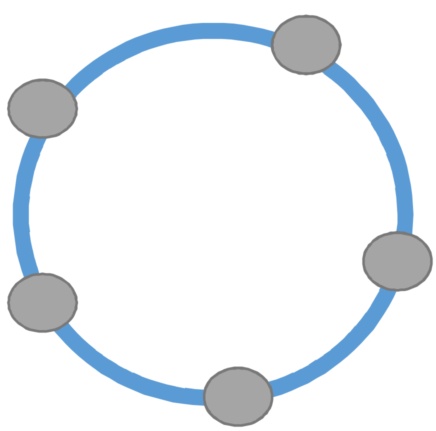

Figure 2. 4. Ring network topology

In star topology, every node connects to center node with point-to-point connection with dedicated link that can be coaxial cable, optical fiber or twisted pair. Central node can be hub, switch or router. Every node has to connect center node where center node behaves as a signal repeater. Star topology is widespread because it provides fast performance, easy installation and management. Easy troubleshooting and adding or removing new nodes is simple. It has also some disadvantages: Installation is expensive, central node represents a single point of failure and performance is sorely based on central node's capacity. It can be a bottleneck for large clusters.

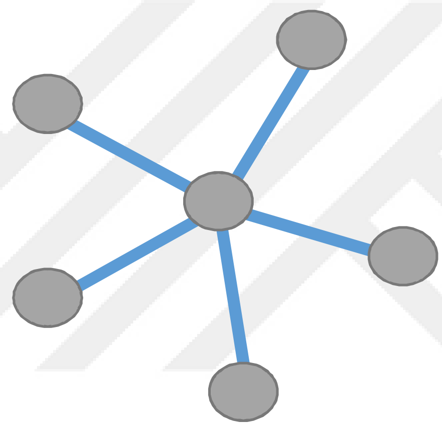

Figure 2. 5. Star network topology

Mesh topology offers two type of mesh network. First one is fully connected mesh network where all nodes are connected with each other and $n(n-1)/2$ physical links are constructed for n nodes. Second one is partially connected mesh network that has some nodes which have point-to-point connection while some others have two or three connections with other nodes. In mesh network, data is transmitted in two ways: routing or flooding. In routing, nodes have some routing method and transmit data according to this method. This method can be information of broken links or shortest distance between nodes. Routing makes mesh network flexible. In flooding, data is transmitted to all nodes so there is no need for some method. It makes network robust but may cause some unwanted load on the network. Mesh network provides robustness, security and easy troubleshooting. However, its cost is expensive and installation is difficult.

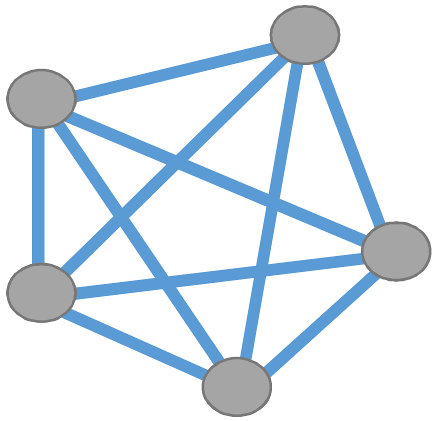

Figure 2. 6. Fully connected mesh network topology

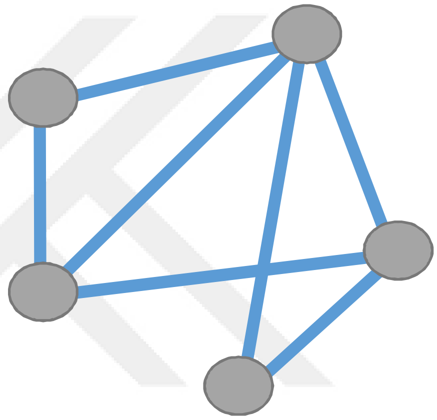

Figure 2. 7. Partially connected mesh network topology

Tree topology is based on a root node and this node connects two or more sub-level nodes where this structure forms a hierarchy. It can be said bus and star networks form a tree network with a root node. Tree topology is also a hybrid topology because it contains two different topologies. It is easy to add nodes and management is not difficult. Error detection is simple. However, the cost of network is not cheap and root node represent a single point of failure. Tree network is ideal if workstation groups are used.

Hybrid topology is formed of two or more topologies where the whole structure does not represent any standard topology. This network inherits advantages and disadvantages of networks that are included. The whole design of hybrid network is complex and expensive. Nevertheless, structure is flexible, scalable and troubleshooting is easy.

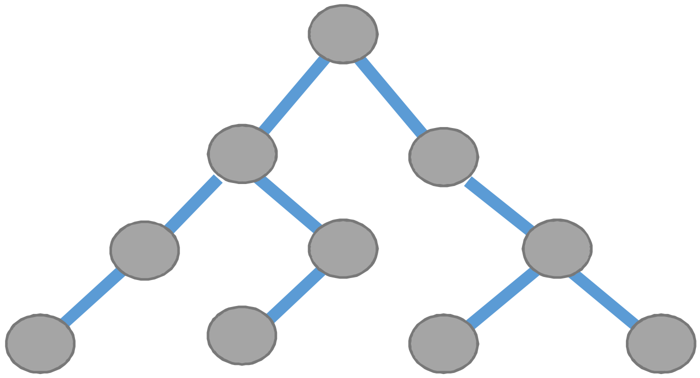

Figure 2. 8. Tree network topology

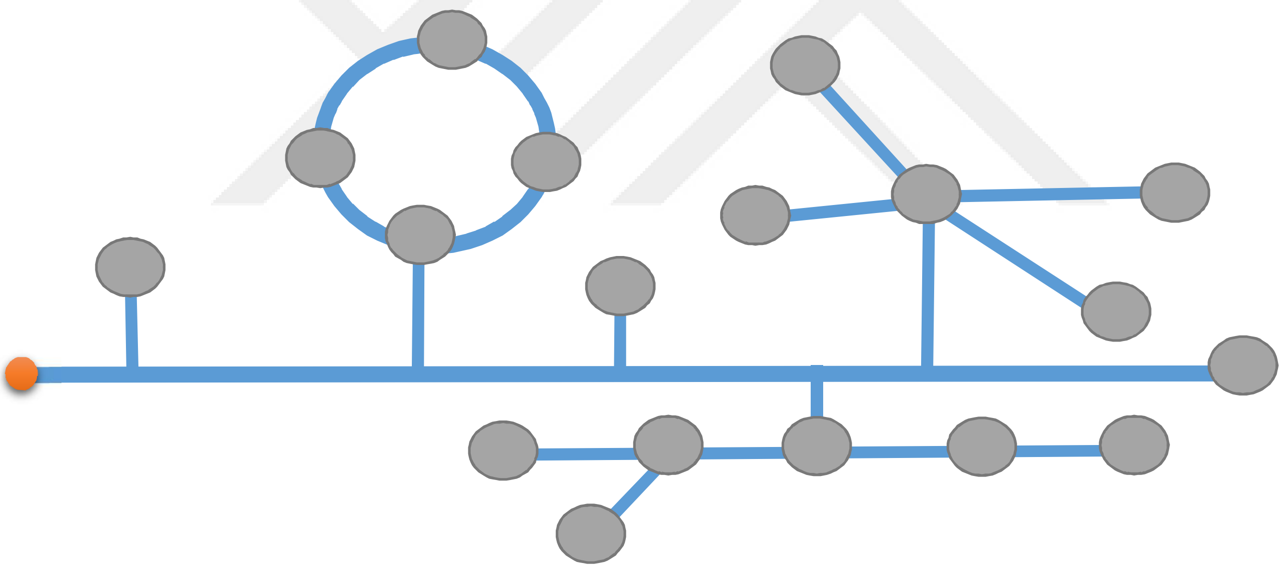

Figure 2. 9. Hybrid network topology

In the light of this information, it is seen that there are several ways to connect sensor nodes to share data. The important thing is the aim and geographical area of WSN.

3. METHOD AND IMPLEMENTATION

3.1 Time Synchronization Algorithm

A sensor node's clock is generally modeled by the following equation:

$$t_a = \alpha_a + \beta_a t_u \tag{3.1}$$

where α_a is the offset, β_a is the drift of the node a and t_u is the universal time [16-19]. In an ideal clock α_a should be zero and β_a should be one where it can behave as l_2 in Figure 3.1. In l_2 θ_2 is equal to 45 degree and t_u and t_{a1} is equal to each other. However, a real clock behaves differently as seen in l_1. There is always an offset difference α_a and a clock rate difference which is β_a that equals to tan(θ). Drift and offset values changes for different nodes and behaves constant for a period of time. Obtaining these values solves the time difference between node a and universal time assuming β_a does not change for a long period.

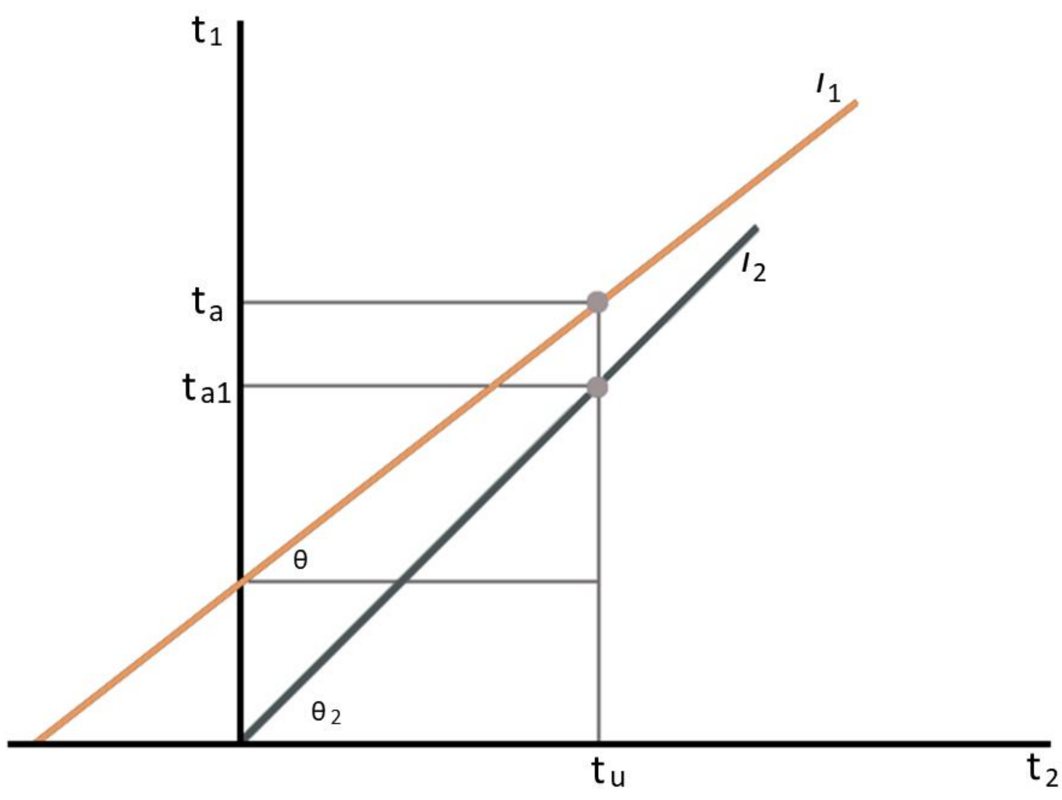

Figure 3. 1 Clock slope

Extending this formula in between two sensor nodes we can have this equation:

$$t_1 = \alpha_{12} + \beta_{12} t_2 \tag{3.2}$$

Parameters t_1 and t_2 are the clocks of node 1 and node 2. α_{12} is the relative offset, β_{12} is the relative drift between two clocks. Assuming clocks of two nodes are well-synchronized relative offset and relative drift should be zero and one respectively. However, since two nodes have generally have different clock rates these ideal values can not be seen. Node 2 and node 1 performs a two way

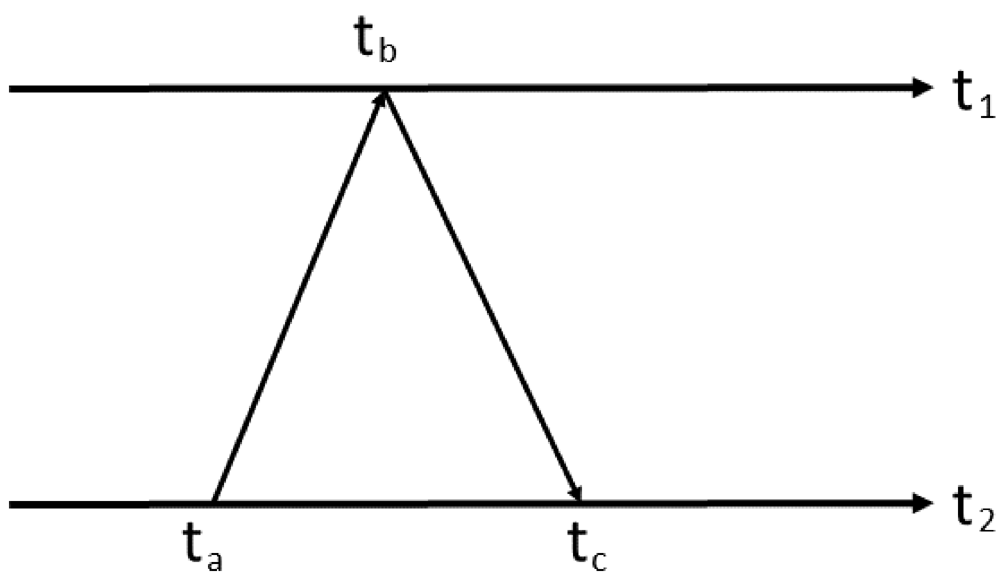

Figure 3. 2 Data-point

handshake to find out relative offset and relative drift. Here t_a is the local time of node 2 that is sent to node 1, when node 1 gets this message it replies with its own clock t_b. Node 2 takes receipt time of reply t_c and three time-stamps (t_a, t_b, t_c) form a data-point. From this data-point node 2 can make estimation on node 1's clock. To improve the estimation several data-points are taken. After enough data-points are taken to find upper and lower bounds of relative drift Equation 3.3 and 3.4 are used.

$$\beta_a(i) = \frac{t_a(i) - t_a(i-1)}{t_b(i) - t_b(i-1)} \tag{3.3}$$

$$\beta_b(i) = \frac{t_c(i) - t_c(i-1)}{t_b(i) - t_b(i-1)} \tag{3.4}$$

If clocks' rate are equal the time duration between consecutive points should be same in $t_a(i) - t_a(i-1)$, $t_b(i) - t_b(i-1)$ and $t_c(i) - t_c(i-1)$. However, since there is a drift β_a and β_b results a value that is around one. By using these values upper and lower bounds of relative offset is found as seen in equation 3.5 and 3.6.

$$\alpha_a(i) = t_a(i) - \beta_a(i)t_b(i) \tag{3.5}$$

$$\alpha_b(i) = t_c(i) - \beta_b(i)t_b(i) \tag{3.6}$$

The variation of relative drifts causes variation on α_a and α_b. Big variations of the relative drift can cause accumulations on the relative offsets. Using upper and lower values, estimation parameters α and β are obtained.

$$\alpha(i) = \frac{\alpha_a(i) + \alpha_b(i)}{2} \tag{3.7}$$

$$\beta(i) = \frac{\beta_a(i) + \beta_b(i)}{2} \tag{3.8}$$

For n consecutive data-points (n-1), different relative offset and relative drift values are estimated. Taking average of these estimation values, we obtain a better result where random delays' effects are eliminated [16].

$$\alpha_{avg} = \frac{1}{n-1} \sum_{j=1}^{n-1} \alpha(j) \tag{3.9}$$

$$\beta_{avg} = \frac{1}{n-1}\sum_{j=1}^{n-1}\beta(j) \quad (3.10)$$

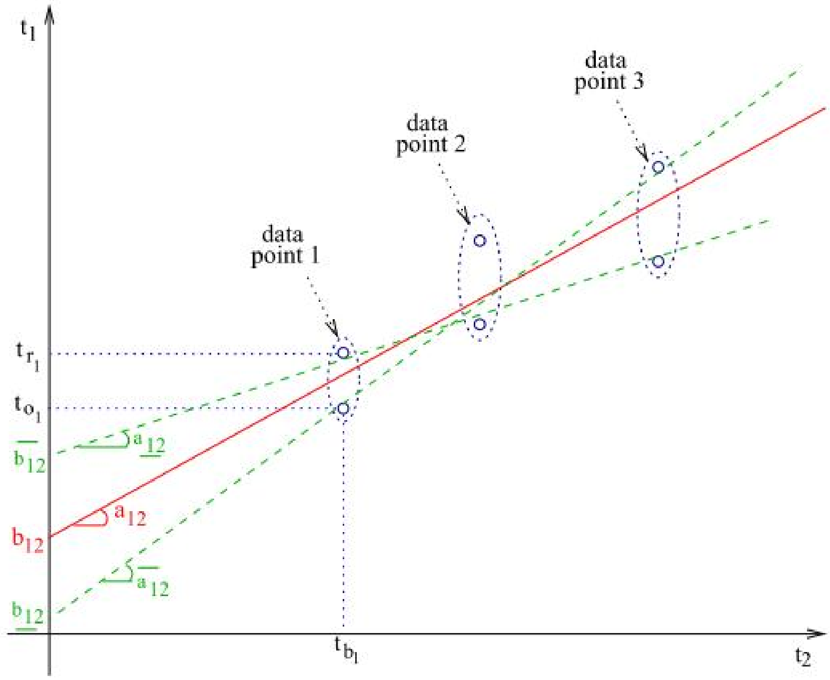

Figure 3. 3 Limitation of the consecutive data-points [10]

In Figure 3.3 the limitations that come with the consecutive data-points are seen. When combining of all point the deciding line should pass between every t_b and t_r pairs. While this limitation method does not exactly applicable it gives general idea how to handle data-points in big picture.

After estimation values are calculated node 2 can update its clock to the reference clock value that should be in synchronization with node 1 by using Equation 3.11.

$$t_{2-estimated} = \frac{t_2 - \alpha_{avg}}{\beta_{avg}} \quad (3.11)$$

3.2 System Model and Method

Wireless sensor network in rural areas has different challenges. In our study, there are sensor nodes that placed on a pipeline where it passes across country. These sensor nodes on the dedicated networks have to make measurements simultaneously. The pipeline often passes rural areas and the nature of rural areas changes depending on geographical location.

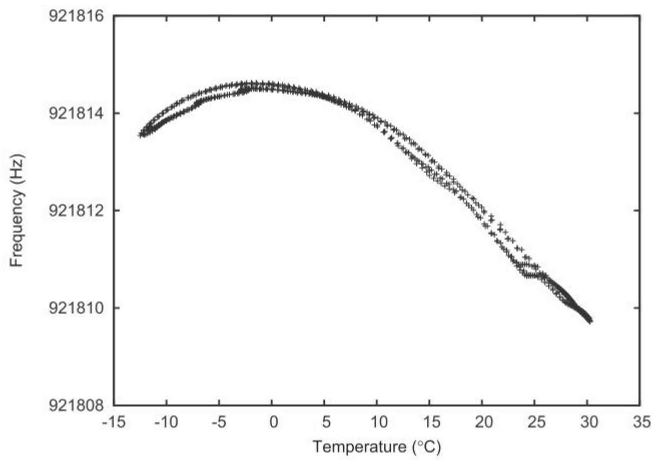

Figure 3. 4 Frequency – temperature of a Mica2 node [23]

Environmental conditions of rural areas like temperature, humidity and RF interference effect the WSN in many ways. [20] Temperature is directly related to the crystal oscillator's stability. In Figure 3.4, it is seen that the temperature affects the elastic constants and dimensions of the oscillator and this leads a change in frequency of resonator. Aging of crystal can cause 1 to 3 part per million (PPM) error where 40 PPM implies a variation of 40 µs per second. An ordinary crystal causes generally 50 PPM. In the experiments, temperature changes' effect is expected to be seen. Humidity both affects temperature stability and RF communication quality on environment. RF interference also has effects on RF communication. These various effects make it harder to provide a good time synchronization on the WSN.

Sensor nodes are distributed in wide geographical area where one-hop distance changes between 1-2 km. A representative distribution of n nodes is seen in Figure 3.5. Although distances between nodes differ, every node can establish one-hop communication with neighbor nodes.

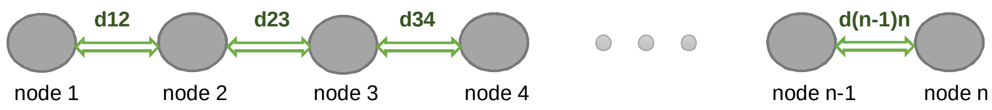

Figure 3. 5 Distribution of nodes

Nodes' distribution seems like a bus topology, however a tree topology is more feasible in this case. Although the structure does not fit in any basic topologies, it is close to a spanning tree where it is a tree topology but also resembles partially connected mesh topology in a way. It is clear that here decision-maker is geographical area to decide topology. There is a root (master) node and other nodes are connected to it in hierarchy in a single branch as seen in Figure 3.6.

A time synchronization method is implemented, where parent-child structure is constructed on whole network from root node to last node. Parent and child nodes perform pairwise synchronization, where child node updates its time with respect to parent node. Every node in the network has a unique ID and these IDs take place on deciding parent-child relation. All nodes except root and last node can be parent and child while root node is always parent and last node is always child.

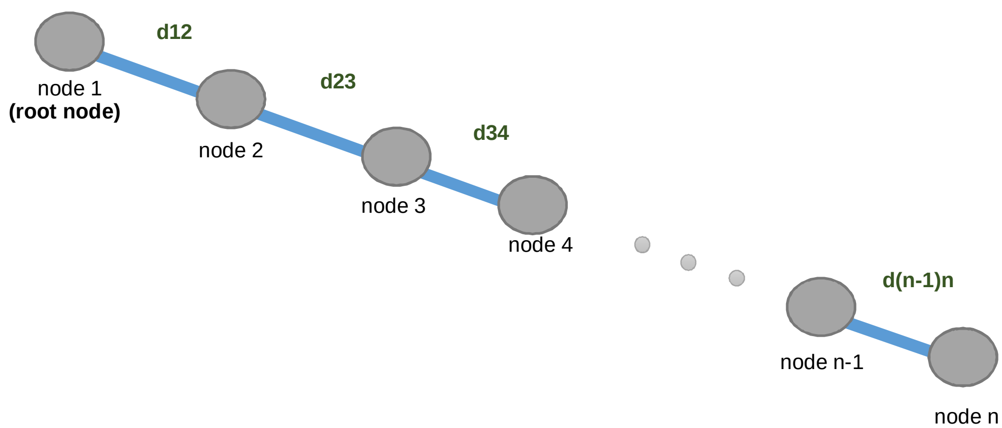

Figure 3. 6 Spanning tree network of nodes

Network's time synchronization process starts with the root node. Root node connects to second node as a parent, child node updates its current time with pairwise synchronization with respect to parent node, and then it disconnects from parent node. After that, previously child node becomes parent node itself where next node becomes child node and same updating process is done. This iterative process will continue until last node updates its current time and disconnects from its parent.

This is how time synchronization at network level is done. Parent-child structure is the place where time synchronization algorithm is applied as explained in section 3.1. Total time synchronization process repeats at certain time intervals and this is called resynchronization interval or resynchronization period.

3.3 System Design

System design is divided into two parts. The first part is hardware design where general hardware structure is explained. The second part is focused on software development.

3.3.1 Hardware Design

Experiment set-ups are combined of three parts. First one is clicker 2 from Mikroeloktranika where Microchip's PIC18F87J50 microcontroller is used for microcontroller. As a communication unit Texas Instruments' (TI) CC1200 RF module is used. Li-ion batteries are used for power.

PIC18F87J50 has 128 KB Flash ROM, 3904 bytes of RAM and integrated full speed USB 2.0 support. It has all main peripherals like UART, SPI, I^2C, DAC and ADC. USB feature is a big advantage; it is both used for programming device and real-time communication with a PC program. Different interfaces of microcontroller makes it easy to communicate with different Medias. SPI is used to communicate with RF Transceiver. Choosing CC1200 provides good configuration on the RF registers since TI's SmartRF Studio helps to run RF chip with best settings for the environment. CC1200 has also some uncertainties about transmitting and receiving timestamps. MCU's external crystal is also not specially chosen so it will show variations in different boards and time synchronization's unknown delays will wave. The general structure is shown at Figure 3.7.

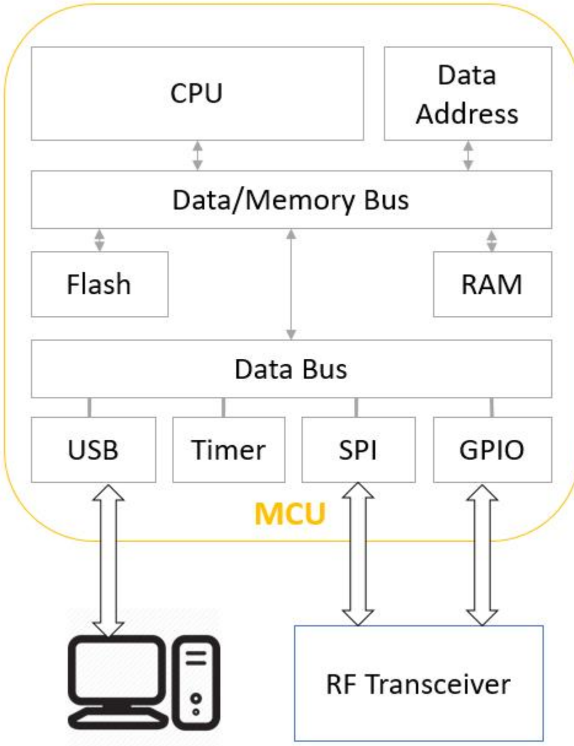

Figure 3. 7 MCU and RF unit

3.3.2 Software Development

In the software development part, MikroC is used as a compiler, which is developed by Mikroeloktronika. MikroC has built-in libraries for various peripherals in PIC and it makes easy to develop new algorithms quickly. It has also some features for real-time data observing which is essential for this study, where experiment data is saved with this way. Bare metal programming is used to implement time synchronization method that is explained in section 3.1. Any abstraction layer based unknown random delays are prevented by using bare metal programming. Data link layer is not used for RF module, to keep non-deterministic delays minimum. RF module access is hold at low level and every step is carefully planned, so the uncertainties at transmitting and receiving are kept at minimum. Time clock with a granularity of 1 ms is maintained using timer interrupt at highest priority interrupt. Resolution of time clock decides also the resolution of error after synchronization which will be also multiplies of 1 ms. Finite state machine is used to build general structure. RF processes and time synchronization calculations are spread through states so that MCU is used efficiently. 8-Bit MCU causes some problems on variable defining. To keep time at high precise it is convenient to use most bits. MikroC has 32-Bit float and integer and this causes some limitations on the calculations. To overcome these problem variables are pushed until overflow occurs thus, maximum precise values for relative offset and drift can be achieved.

3.4 Experimental Setup

Experimental setup is consisted of two computers and six sensor nodes to demonstrate given method in section 3.2. The general setup is shown at Figure 3.8 where test procedure is seen at Figure 3.9. First, the computers' clock should be synchronized. To do this, a program called

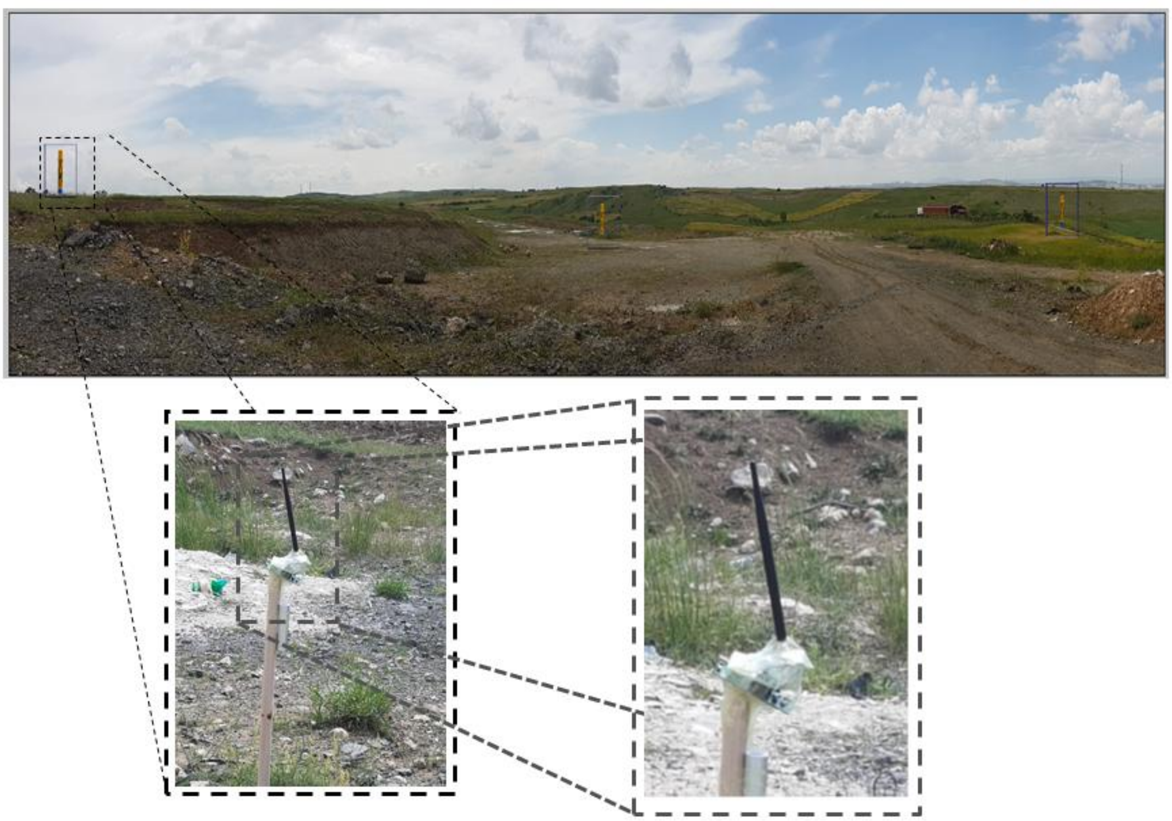

Figure 3. 8 WSN Environment

netTime is used which uses Simple Network Time Protocol (SNTP) [21]. By using this program on two computer, their time is synchronized with millisecond precision. Every sensor node can connect to computer and their clock can be seen and saved by a program called RF Timer, which is created for this study.

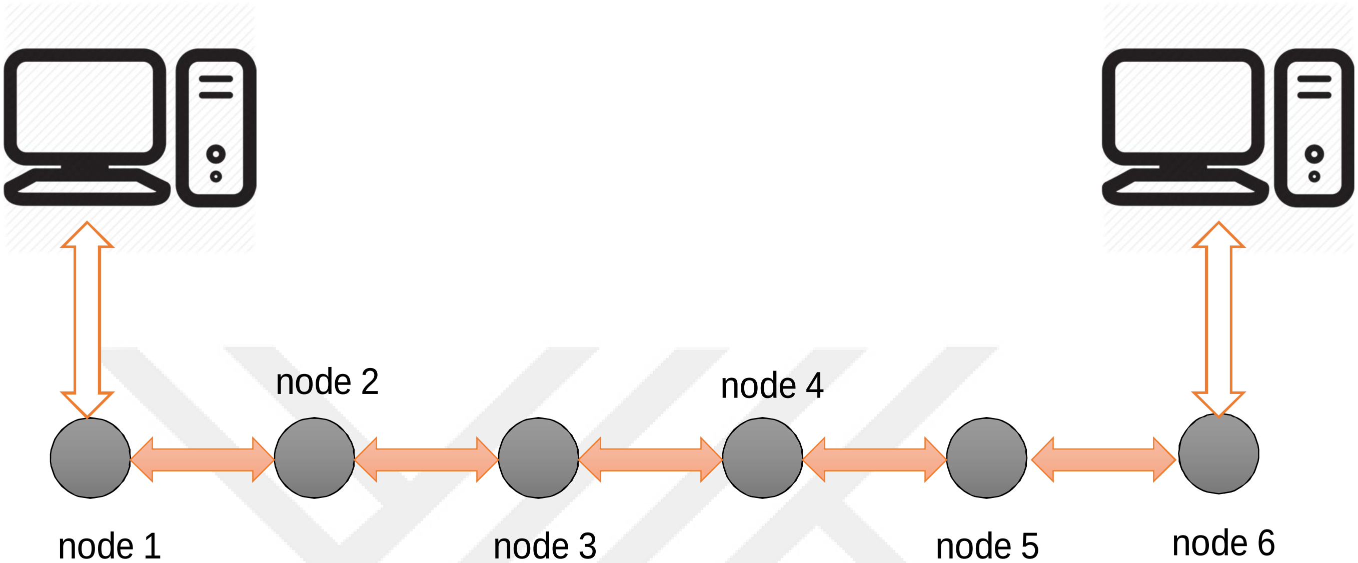

Figure 3. 9 Experimental setup

RF Timer program is connected to nodes via USB. This program reads computers local time and at every second (exactly start of the second) asks node's local time (in millisecond) and save node's response to a text file.

When the computers, which are connected to node 1 and 6 are synchronized, they can ask node's local time at the same instant. RF Timer can send messages from node 1 to node 6 to see connection's situation and if link is connected. Usage of RF Timer is shown at figure 3.11.

Test procedure proceeds with the following steps:

- A link is established with different distances between nodes with six nodes.
- 20-30 instances of time synchronization process is started and saved with environment's temperature.
- Two of nodes are heated by using a heater and again 20 - 30 instances of time synchronization process is started and saved while other nodes' temperature is equal to environment's temperature.

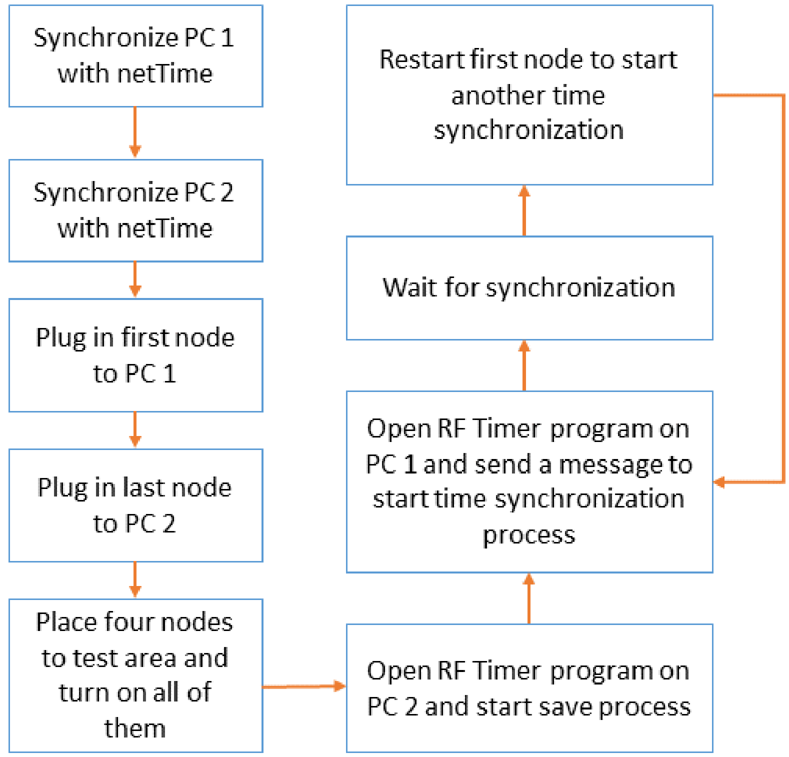

Figure 3. 10 Test procedure

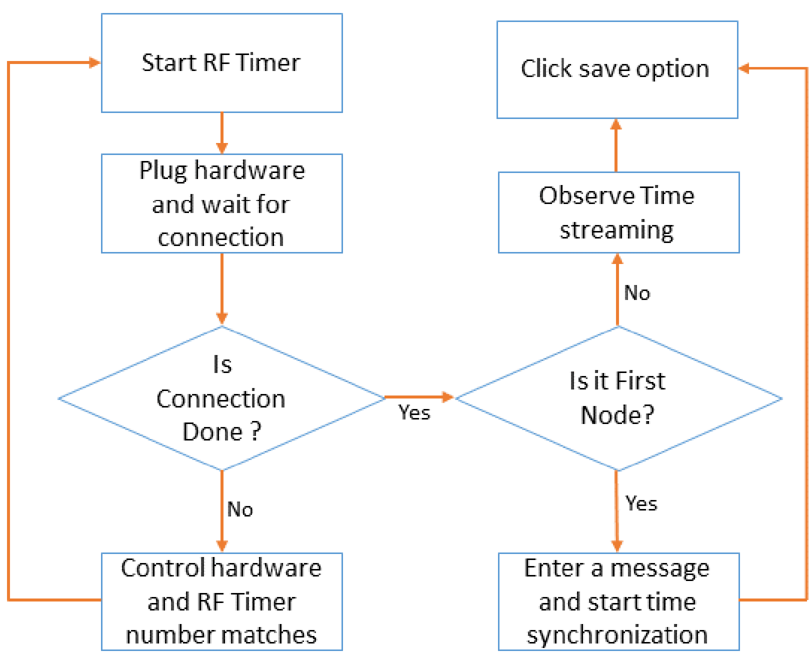

Figure 3. 11 RF Timer Usage

In Figure 3.12, experimental setup is seen with nodes. Every node is connected to approximately 1 meter long stick to keep signal health good. Rural area scenes is seen in Figure 3.13.

Figure 3. 12 Sensor nodes

Figure 3. 13 Rural area of landscapes

4. EXPERIMENTS AND RESULTS

There are four experiments. They are divided into two types as indoor and outdoor. First two experiments are performed at a rural area where different outdoor effects take place like RF interference, fluctuant temperature, humidity, obstructed line of sight links and various RF reflector materials. Time synchronization algorithm's performance is tested with different hours of the day where these conditions also vary with time.

Last two experiments take place in a closed area where some sensor nodes have only walls as an obstruction and some has line of sight links. In this area, stable temperature and humidity are kept with an air conditioner and the time synchronization algorithm performance is observed. Than some of nodes' temperature is increased by an heater and performance is observed once again. Various problems are occurred during these tests, thus measurement numbers differ for every experiment.

In these experiments, performances of indoor and outdoor are expected to be different. Temperature effect is also one of expected distinguishing feature. To perform data analysis mean value and standard deviation are applied on the experiment results. Mean value gives the most common result of the experiment where standard deviation talks about behavior of the data by looking how spread out a data set is. For a random variable vector V made up of N values, mean (μ) and standard deviation (σ^2) can be calculated by using the following equations:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} V_i \qquad 4.1$$

$$\sigma^2 = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} |V_i - \mu|^2} \qquad 4.2$$

In the experiment 1 that is seen in Figure 4.1, six nodes are placed on outdoor environment with distances approximately 150 m. The experiment is started at 23℃ and the temperature is raised to 27℃ for all nodes. The rise at temperature is natural; it is not obtained from a heater or suchlike things. Environment of the nodes are consisted of empty rural areas. Experiment is started at morning so RF interference is not so high. The nodes are attached to a wooden stick, which is 1 m

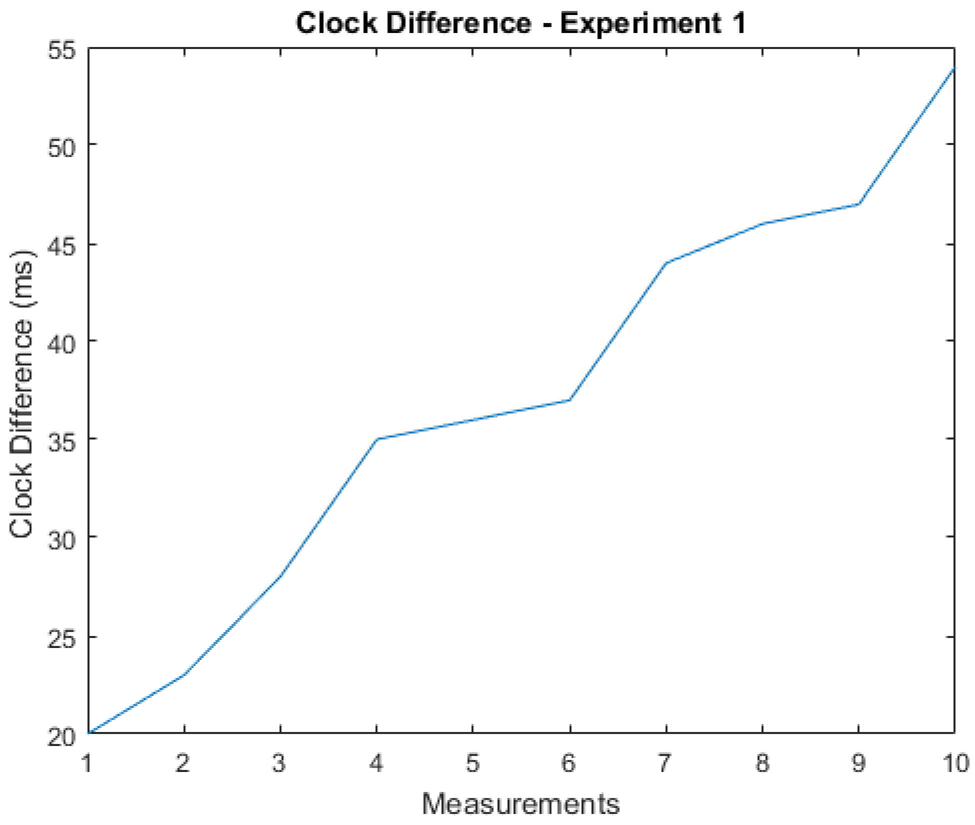

Figure 4. 1 Experiment 1

long. The results vary from 20 to 54 ms, long distances and fluctuation on temperature cause this outcome.

Second experiment is also done at the same place with experiment 1, so the environmental features are the same. However, the experiment takes place at noon so RF interference is different. In addition, the temperature is changed from 27℃ to 33℃ with naturally by environment. Results of experiment 2 starts from 37 ms and ends at 104 ms. In this experiment a smooth start is observed but than a sudden increase is seen. The total temperature difference is a decisive effect on these

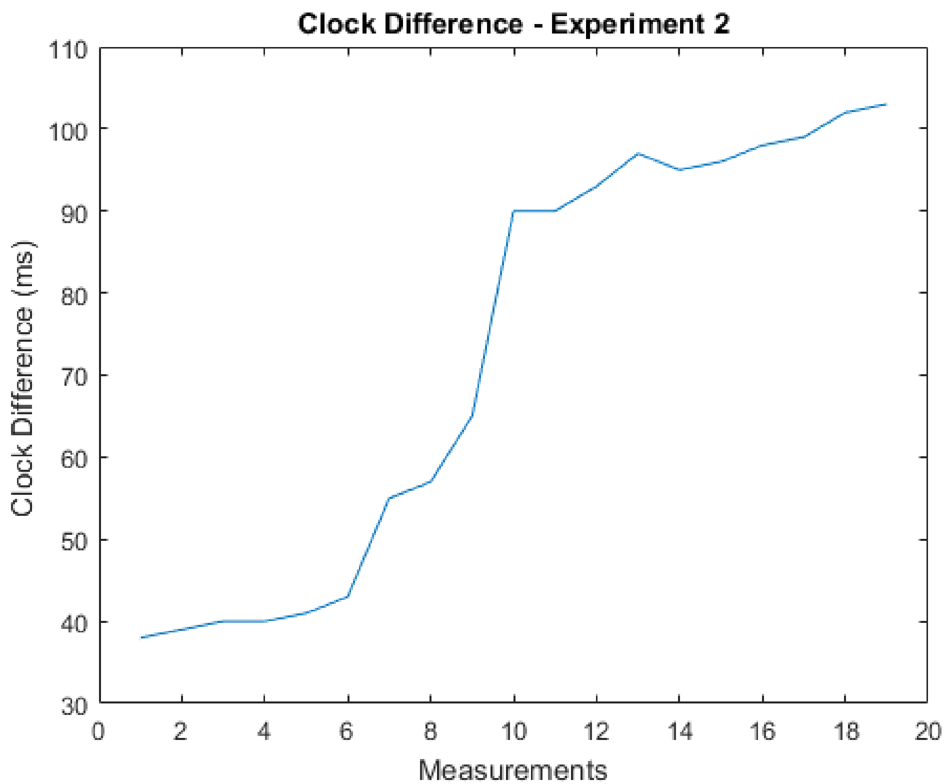

Figure 4. 2 Experiment 2

results where they are spread widely. It has also different first synchronization time values with respect to experiment 1 as they are 20 ms and 37 ms. Experiment 1 and 2 have similar results as outdoor tests.

In Experiment 3, nodes are placed in an indoor environment where many walls are stayed between some nodes. Temperature is stable around 24℃. Experiment time is done at noon. Nodes are attached randomly in rooms the distances between them differ in range of 20-40 m. The environment has various people where the density changes over time. The results vary from 8 to 0 ms, indoor results are very different outdoor result at first glance when the first synchronization time is focused. The results behave as a decreasing linear smooth data cloud.

Experiment 4 takes place at the same environment with experiment 3. The time is afternoon. Two of nodes are heated from 24℃ to 35℃ others are remained around 24℃. The distance of the nodes remain same as the experiment 3. First synchronization time is changed as 11 ms and the results

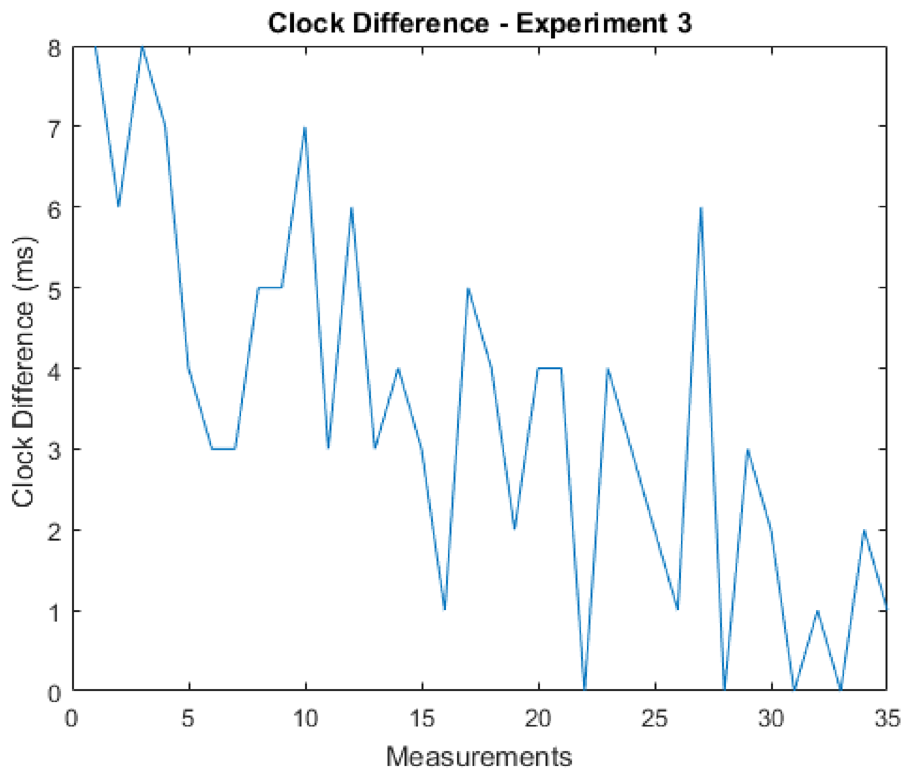

Figure 4. 3 Experiment 3

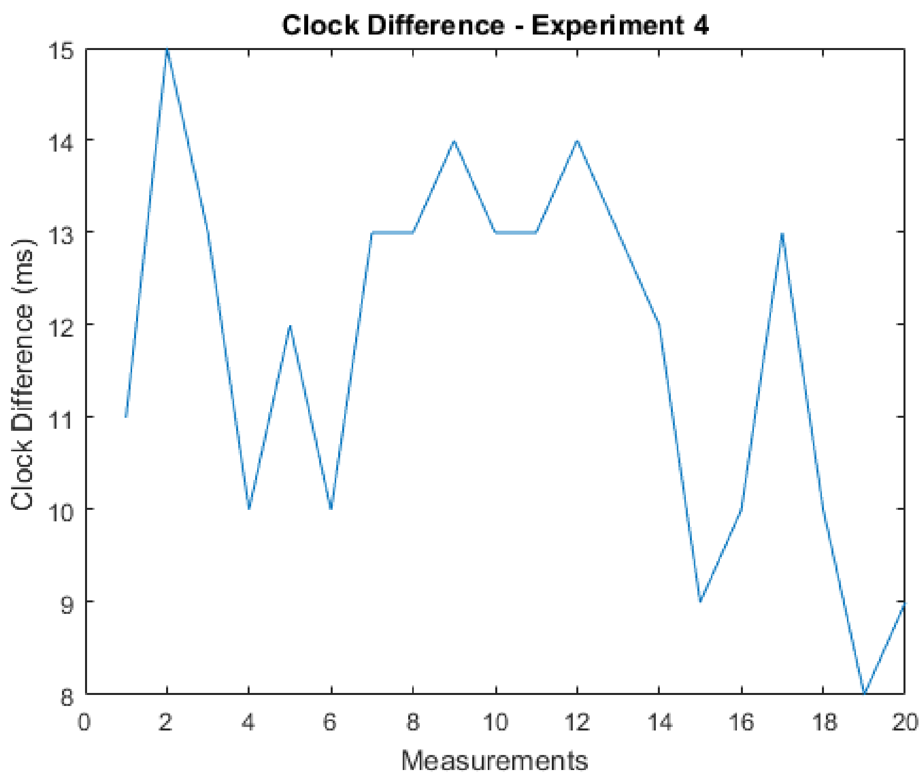

Figure 4. 4 Experiment 4

are spread between 15 – 8 ms. The decreasing data pattern in experiment 1 has slowed down in experiment 2.

In Table 4.1, mean and standard deviation values are seen. An analysis can be done by combining

Experiment Number	Mean (ms)	Standard Deviation(ms)
1	37.00	11.00
2	72.68	26.59
3	3.42	2.27
4	11.75	1.97

Table 4. 1. Mean & standard deviation values for experiments.

these values with the experiment figures. Experiment 1 and 2 have high first synchronization time where second one has highest due to high temperature. Since second experiment has twofold measurement number, mean and standard deviation is also doubled. However, standard deviation is clearly more than twofold; this shows us the high temperature change in experiment 2 has an effect on standard deviation. Experiment 3 and 4 have relatively small first synchronization time and a non-spreading behavior of values when we look at standard deviation results. On the other hand, two heated sensor nodes directly have an impact on time synchronization and mean value takes a big leap. The change of standard deviation in experiment 4 is also the evidence of the influence on the time synchronization, yet it is not high as expected.

5. CONCLUSION

In this study, time synchronization on a wireless sensor network is aimed with different aspects. Various time synchronization protocols are introduced and their pros and cons are studied. A wireless sensor network structure is researched with both hardware and topological aspects. The natural gas pipeline makes headway to rural areas with directly passing into them or by-passing them. When this behavior of pipeline is analyzed through topological view, line topology and tree topology are found favorable. A network with hierarchical nodes that communicate with multi-hop can be established. On the other side, we see that multi-hop spanning tree wireless sensor networks are not tried on rural areas. Herewith, given these realities, a simple linear spanning tree wireless sensor network is designed. Pair-wise communication availability and scalability of this network make this process simple and practical.

A basic sensor node is established and a lightweight time synchronization algorithm is developed on these sensor nodes where they formed a linear spanning tree WSN. The important point is bare metal programming is done on the embedded system and while every step is planned, the complexity is avoided to the greatest extent possible. One millisecond resolution is chosen for the clock due to fact that hardware is not stable under this value and expected synchronization time can be achieved with this resolution. The algorithm is designed generic so if the number of nodes increases it can easily support that with some basic changes on it, however there will be a maximum node number where performance of the algorithm decides. Algorithmically low complex structure leaves microcontroller extra process power to handle ambiguous problems on the rural area. On the other hand, low complexity means low power usage since power takes an essential part in wireless sensor networks.

Temperature change on a crystal oscillator is always vital for time focused embedded systems and wireless sensor networks rise on these systems. In this study, sensor nodes use ordinary crystals with average error values. Development of the algorithm with these ordinary crystals makes focus point directly algorithm. It says us with any hardware a good time synchronization can be achieved. If a dedicated crystal oscillator will be used with the algorithm, performance of the algorithm will be naturally better.

Finally, experiments are done on outdoor and indoor areas to test linear spanning tree WSN and temperature effects on time synchronization. Linear spanning tree WSN is successfully established and the synchronization times are resulted within the limits of expected time interval value. Results show that outdoor and indoor performances of time synchronization algorithm are clearly different from each other; however, it is still good enough to provide the expected value. Although every node has an identical crystal oscillator, every crystal oscillator has unique skew behavior and its response to temperature variation is unique too [20]. An example of a single different crystal oscillator can be seen in Figure 3.4. This unique change in crystal oscillator frequency does not change only local time of sensor node but also important time delays in time synchronization algorithm. Pre-calculated time delays in RF communication are severely affected from this change and the communication timings directly affects time synchronization algorithm [22]. Indoor and outdoor results are obviously far apart from each other, the outdoor performance of the algorithm give some flaw signs in the algorithm. Although, the performance is good enough there may be unwanted results under heavy environmental conditions. Last experiments give literally high mean difference but small standard deviation results; only two nodes' temperature change does not affect general synchronization values as much as outdoor effects but it is still a disturbance that needs to be overcome. While mean value is high, the standard deviation is small. In some cases, the result can be better like this where similar experiment is done at [10] and good result is seen.

The objective of the study is to use this algorithm at outdoor and the results are not bad but it can be improved by finding optimum resynchronization interval and forming a temperature based clock formula for ordinary crystal oscillators.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. Wireless Sensor Networks: A Survey. Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002.

[2] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: A Survey," Network, IEEE Vol. 18, No. 4, pp. 45–50, 2004.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in 5th USENIX Symp. Operating System Design and Implementation (OSDI'02), Dec. 2002.

[4] D. Cox, E. Jovanov, A. Milenkovic, "Time Synchronization for ZigBee Networks," in Proc. of the 37th IEEE Southeastern Symposium on System Theory (SSST'05), Tuskegee, AL, March 2005, pp. 135-138.

[5] S. Ping, Delay measurement time synchronization for wireless sensor networks, Intel Research, IRB-TR-03-013, June 2003.

[6] L. M. He, "Time Synchronization for Wireless Sensor Networks," *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, Daegu, 2009, pp. 438-443.

[7] Jana van Greunen , Jan Rabaey, Lightweight time synchronization for sensor networks, Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, September 19-19, 2003, San Diego, CA, USA

[8] J. Wu, L. Zhang, Y. Bai, Y. Sun, Cluster-based consensus time synchronization for wireless sensor networks, IEEE Sens. J., 15 (2015) 1404-1413.

[9] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," in *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 384-397, April 2005.

[10] Yoon, S., Veerarittiphan, C., and Sichitiu, M. L. 2007. Tiny-Sync: Tight time synchronization for wireless sensor networks. ACM Trans. Sens. Netw. 3, 2, Article 8 (June 2007), 34 pages.

[11] E. Garone, A. Gasparri and F. Lamonaca, "Clock synchronization protocol for wireless sensor networks with bounded communication delays", Automatica, Volume 59, 2015, Pages 60-72,

[12] M. L. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, New Orleans, LA, USA, 2003, pp. 1266-1273 vol.2.

[13] J. Chen, Q. Yu, Y. Zhang, H. H. Chen and Y. Sun, "Feedback-Based Clock Synchronization in Wireless Sensor Networks: A Control Theoretic Approach," in *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2963-2973, July 2010.

[14] I. F. Akyıldız and M. Vuran, Wireless Sensor Networks. Wiley, 2009.

[15] F. Bonavolontà, A. Tedesco, R. S. L. Moriello and A. Tufano, "Enabling wireless technologies for industry 4.0: State of the art," *2017 IEEE International Workshop on Measurement and Networking (M&N)*, Naples, 2017, pp. 1-5.

[16] G. Bam, E. Dilcan, B. Dogan, B. Dinc and B. Tavli, "DLWTS: Distributed Light Weight Time Synchronization for Wireless Sensor Networks," *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nusa Dua, 2015, pp. 447-450.

[17] J. He, P. Cheng, L. Shi, J. Chen and Y. Sun, "Time Synchronization in WSNs: A Maximum-Value-Based Consensus Approach," in *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 660-675, March 2014.

[18] Amulya Ratna Swain, R.C. Hansdah, A model for the classification and survey of clock synchronization protocols in WSNs, Ad Hoc Networks, Volume 27, 2015, Pages 219-241,

[19] Djamel Djenouri, Nassima Merabtine, Fatma Zohra Mekahlia, Messaoud Doudou,

Fast distributed multi-hop relative time synchronization protocol and estimators for wireless sensor networks, Ad Hoc Networks, Volume 11, Issue 8, 2013, Pages 2329-2344,

[20] F. L. Walls and J. J. Gagnepain, "Environmental sensitivities of quartz oscillators," in *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 39, no. 2, pp. 241-249, March 1992.

[21] M. Ussoli and G. Prytz, "SNTP time synchronization accuracy measurements," *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Cagliari, 2013, pp. 1-4

[22] A. Marco, R. Casas, J. L. Sevillano Ramos, V. Coarasa, A. Asensio and M. S. Obaidat, "Synchronization of multihop wireless sensor networks at the application layer," in *IEEE Wireless Communications*, vol. 18, no. 1, pp. 82-88, February 2011.

[23] C. Lenzen, P. Sommer, and R. Wattenhofer. “Optimal clock synchronization in networks.” In SenSys, 2009.