공학석사학위논문

A Time Synchronization Protocol for TDMA Based Wireless Sensor Networks

TDMA 기반의 무선 센서 네트워크를 위한 시간 동기화 프로토콜

2013년 8월

서울대학교 대학원

전기 컴퓨터 공학부

Eric Yip

A Time Synchronization Protocol for TDMA Based Wireless Sensor Networks

TDMA 기반의 무선 센서 네트워크를 위한 시간 동기화 프로토콜

지도교수 이 정 우

이 논문을 공학석사 학위논문으로 제출함

2013년 5월

서울대학교 대학원

전기 컴퓨터 공학부

Eric Yip

Eric Yip의 공학석사 학위논문을 인준함

2013년 6월

위 원 장 : ____________________

부위원장 : ____________________

위 원 : ____________________

Abstract

A Time Synchronization Protocol for TDMA Based Wireless Sensor Networks

서울대학교 대학원
전기컴퓨터공학부
Eric Yip

There has been much interest in wireless sensor networks recently, due to their diverse range of possible applications. Although there have been much research in MAC layer protocols for wireless sensor networks, these works are mainly focussed on the power savings and efficiencies of the protocols. For sensor networks which are in-situ and do not require much flexibility, such as a battery management system, energy is not always the most important factor, but rather reliability and scalability (where sensing periods are known). As such, a traditional TDMA protocol can be considered as a good option.

Time synchronization in wireless sensor networks have also been considered by many academics, but work related to time synchronization in TDMA networks have been much less popular. In this thesis, a time synchronization protocol for TDMA based wireless sensor networks is proposed, Propagating Chain Time Synchronization.

Propagating Chain Time Synchronization is a novel protocol for synchronizing TDMA based networks. The scheme achieves improved synchronization errors compared to traditional beacon synchronization methods, through skew correction estimated from chained two-way message exchanges, which employ piggybacking and overhearing.

Keywords: wireless sensor networks, WSNs, time synchronization, TDMA

Student Number: 2011-24082

Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) have become a hot topic of discussion in recent years, not only in academia but also in industry due to their wide range of possible applications. Whereas traditional sensor networks are deployed using wired networks, the application of a wireless sensor network allows for cost savings in terms of deployment resources (wired connections between sensor nodes), as well as increased flexibility in terms of the network topology. Additionally, many applications of wireless sensor networks cannot even be replaced by the traditional wired networks due to the actual sensing environment (most specifically remote networked sensing where devices nodes may be mobile and operating in remote locations).

Network sensing can be divided into two brief categories: embedded (in-situ) networked sensing, and remote networked sensing.

- Embedded sensing involves sense data which is at the sensor location, where sensor nodes are not typically mobile. Example sense data may include seismology, temperature, pressure and voltage.

- Remote networked sensing involves data which is at a remote location, which may include cameras, vehicle tracking, radar and GPS.

Although it seems that all embedded wireless sensor networks can be replaced by traditional wired networks, this is not always the case. The deployment location and method of sensor nodes can be a limitation in using wired or wireless networks. In certain applications where even node deployment itself is dangerous, difficult, and expensive enough, wire installation for a wired sensor network would seem absolutely unfeasible.

1.1.1 Challenges in Designing Wireless Sensor Networks

Academic interest in wireless sensor networks arose not only due to the wide range of applications achievable by these networks, but also due to many difficult challenges concerned with using such networks. Since different networks will have different requirements with regards to factors such as power, sensing period, cost etc., the challenges in designing for a wireless sensor network will certainly differ depending on the application. In general, the challenges in designing for a wireless sensor network include:

<u>Power</u>

In many networks sensor nodes are remotely powered either by batteries or by some other more sustainable source such as renewable energy (e.g. solar power). Power management is a key issue in many wireless sensor networks, since maintenance costs can be expensive, minimal power consumption is preferable, and there exists many MAC protocols which have been developed in an attempt to lower the power consumption of nodes. The important idea in power management is exactly when energy should be used by nodes, whether to receive, send, or manage data.

Topology

The extreme flexibility of different wireless sensor applications results in many different possible topologies. Figure 1-1 shows the different possible network topology types; for sensor networks where the main objective is to sense and gather remote data, topologies which support a coordinator node are most frequently used (star and tree topologies).

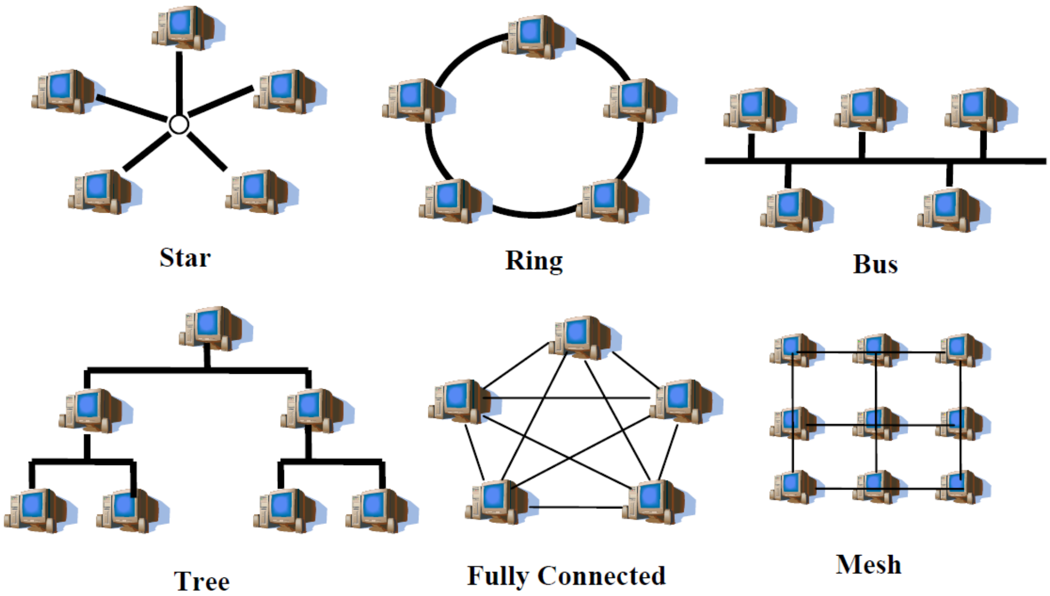

Figure 1-1 - different topologies of sensor networks [1]

Scalability

Closely related to the topology of sensor networks is their scalability. Without the need for wire installations for each node, the limitations on scalability are due to bandwidth and protocol issues (which include time synchronization).

Reliability

Depending on the type of wireless sensor network, the reliability of nodes and their data transmission varies greatly. For remote networked sensing networks, where there may be redundancy in the number of nodes, reliability of data transfer may not be of vital importance, however, for a sensor network which may be health related, the consequence of an unreliable network could be detrimental.

In order to address the challenges listed above, novel protocols and algorithms have been developed to satisfy the unique resource constraints and application requirements of sensor networks [1]. Table 1 gives a qualitative overview of MAC protocols for sensor networks; interested readers are referred to the relevant reference.

MAC protocol	**Channel access mode**	**Sensor network specifics**	**Power conservation**
SMACS and EAR	Fixed allocation of duplex time slots at fixed frequency	Exploitation of large available bandwidth compared to sensor data rate	Random wake up during setup and turning radio off while idle
Hybrid TDMA/FDMA	Centralized frequency and time division	Optimum number of channels calculated for minimum system energy	Hardware-based approach for system energy minimization
CSMA-based	Contention-based random access	Application phase shift and pretransmit delay	Constant listening time for energy efficiency

Table 1 - reconstructed from [1]

More specifically, different MAC protocols have been proposed specifically for wireless sensor networks, including: Sensor-MAC [2], WiseMAC [3], Traffic-Adaptive MAC Protocol (TRAMA [4]), SIFT [5], and DMAC [6]. All of these protocols were designed with energy conservation as one of the top priority requirements, employing techniques such as periodic sleep-listen schedules based on virtual clusters; other ideas include sleep scheduling in order to minimize overhearing, and preamble sampling in order to decrease idle listening. In general, there is a trade-off between latency and energy consumption; almost all novel protocols employ the use of contention based channel access, such as CSMA, as this allows for more flexible node sleep times when there is no data to be sent. However, as will be discussed in section 1.2, some wireless sensor networks which require regular sensing intervals should benefit more from a TDMA based protocol, eliminating collisions due to contention based channel access, and achieving higher latencies with regular sensing.

	Time Synch. Needed	*Comm. Pattern Support*	*Type*	*Adaptivity to Changes*
S-MAC / T-MAC / DSMAC	No	All	CSMA	Good
WiseMAC	No	All	np-CSMA	Good
TRAMA	Yes	All	TDMA / CSMA	Good
SIFT	No	All	CSMA/CA	Good
DMAC	Yes	Convergecast	TDMA / Slotted Aloha	Weak

Table 2 - comparison of MAC protocols [7]

1.2 Thesis Motivation

1.2.1 Wireless Sensor Networks in Battery Management Systems

The main motivation of this thesis is the application of a wireless sensor network for a battery management system. Battery management systems can in itself be applicable to several different environments, such as a smart-grid power system (Figure 1-2), or an in-car battery system. The principle idea is to replace traditional wired systems by wireless systems in order to reduce installation and maintenance costs, as well as allowing for more flexibility in the network architecture. Through a wireless sensor network, various data regarding battery cells' states can be sensed (such as voltage and temperature) and sent to a coordinator, after which certain actions and measures can be taken, such as active cell voltage balancing.

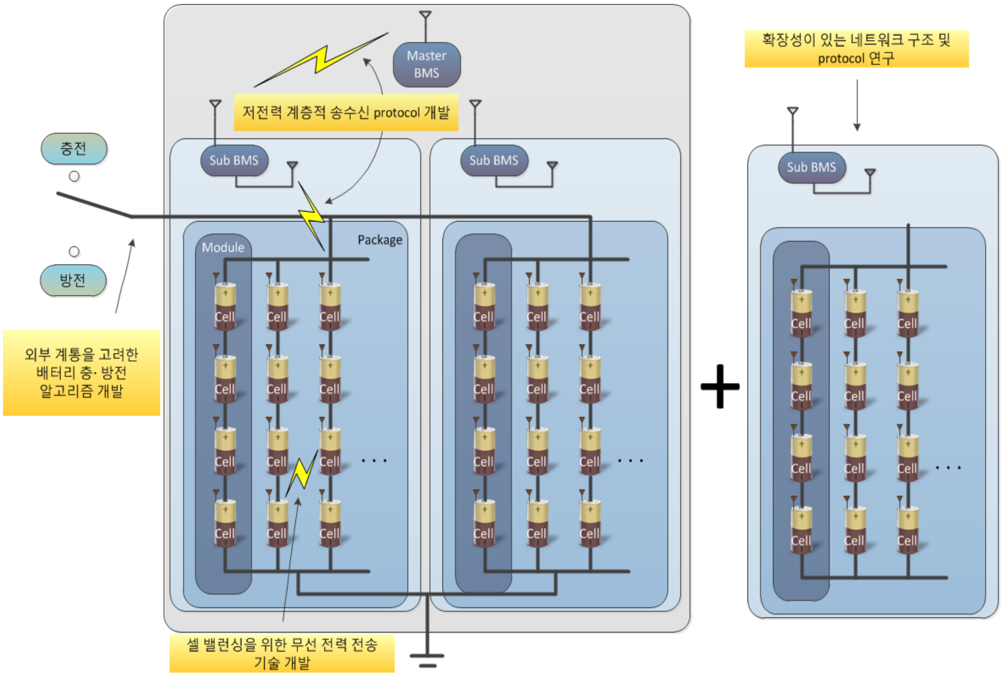

Figure 1-2 - illustrating a wireless sensor network in a smart-grid power system

Requirements

A battery management system can be regarded as an embedded, in-situ wireless sensor network where energy efficiency is not considered a top priority design criteria, since nodes will have almost unlimited power from batteries. Instead, for a battery management system where voltages and other data are sensed periodically, a scalable, reliable and throughput efficient protocol is best suited to the cause.

A TDMA protocol which has the natural advantage of a collision-free medium access provides good latency for periodic sensing networks, as well as throughput fairness among all sensor nodes, however, problems may include clock drift problems and adaptation to topology changes [7]. [1] also emphasizes that although a pure TDMA scheme minimizes the transmit on-time (thus saving energy), it is not always preferred due to the associated time synchronization costs.

In selecting a protocol for a battery management system which has a given sensing period requirement, and where reliability is of greater importance than energy conservation, non-contention based TDMA can be considered as a suitable implementation. Furthermore, throughput fairness is achieved among all sensor nodes, and the immobile, static and predictable nature of a battery management system does not require a highly adaptable protocol.

The important issue which this thesis attempts to resolve is the clock drift problem associated with using TDMA protocols: by introducing a novel time synchronization protocol applied to TDMA protocols, it is possible to achieve better synchronization without the use of specific time synchronization phases, effectively reducing time synchronization costs.

Chapter 2

2. Time Synchronization

2.1 Overview

For sensor networks, clock synchronization is an important service, as in any distributing computer system. For a network which has the possibility to sense information on a broad scale geographically (in terms of location), time synchronization can be used to integrate data and fuse sensor readings with regards to time. Perhaps more importantly, time synchronization is also used for medium access scheduling in TDMA, in fact, time synchronization is possibly the factor which limits the use of TDMA in supporting many devices.

Typical clocks in sensor devices consist of quartz-stabilized oscillators as well as a counter which is decremented with every oscillation of the quartz crystal. When the counter reaches zero, an interrupt is generated and its value is reset to the original value. Each interrupt then increments another counter called a software clock. It is this software clock which can be read by applications using the application programming interface (API). The local time given by the software clock $C(t)$ can be regarded as a function of the real time t (the coordinated universal time UTC). Before introducing in more detail about different time synchronization techniques and models, the basic clock parameters are defined.

Clock parameters and terms [8]

Clock rate: the frequency at which a clock progresses (by referencing the rate to real time, a clock rate of 1 represents real time).

Clock offset: the difference between the local times of two nodes (in seconds).

Clock drift: the ratio between the clock rates of two nodes.

Clock skew: the difference between the clock rates of two nodes.

Synchronize clock time: to set nodes' local clock times at a particular instance of time to be exactly the same such that their clock offset is zero.

Synchronize clock rate: to adjust nodes' clocks to run at the same frequency.

2.2 Models of Clock Synchronization

Time synchronization requirements can be split into many different types of models, each with its own usage [9]:

Chronology of events: in many instances precise real time values may not be important, but rather the chronology and ordering of events. It may be sufficient for a system to simply determine a correct sequence of events, stating whether certain events have happened before or after others. This is the simplest model of synchronization which requires the least resources; achieving synchronization on other models noted below automatically includes synchronization for the chronology of events.

Relative notion of time: this is the synchronization of nodes not with real time, but with some other logical notion – perhaps some unit after an event. This is mostly sufficient for a variety of applications, and is also achieved by synchronizing with regards to real time.

Relative clocks: in this model nodes within a sensor network are synchronized with any other node in the network, but with a time which may be totally different from real-time UTC, essentially the relative notion of time within the sensor network.

Global clock: here all nodes are synchronized with respect to a precise global time, the coordinated universal time UTC, most likely through a reference node in the network.

For wireless sensor networks and, particularly those which utilise TDMA MAC protocols, the relative clock model of time synchronization is the most commonly used model. The rest of this thesis will focus on the achieving time synchronization applied to TDMA wireless sensor networks according to the relative clock synchronization model.

2.2.1 Typical Synchronization Errors

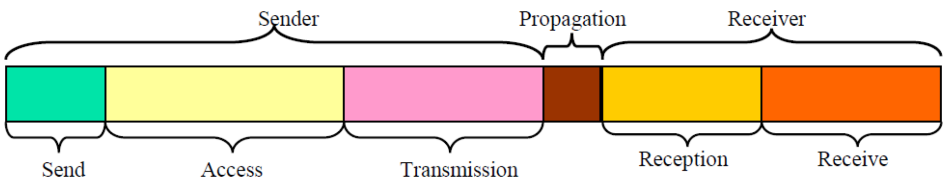

Figure 2-1 - the different types of synchronization error [10]

It is well accepted that the sources of synchronization error can be categorized into the different types as shown in Figure 2-1.

Send time: the time actually spent to construct the packet at the application layer, including the time taken for packet to reach the MAC layer to the application layer.

Access time: the time spent waiting after reaching the MAC layer in order to access the channel; typically highly variable especially for contention based access protocols.

Transmission time: the time when a packet is transmitted bit by bit at the physical layer.

Propagation time: the time taken for the packet to traverse the wireless link, between the sender and the receiver.

Reception time: the time taken in receiving the bits from the physical channel and passing them onto the MAC layer.

Receive time: finally the time taken to aggregate the bits into a packet which is then passed on to the application layer.

Typically the most nondeterministic and difficult to estimate are the send and access times, since they can be highly variable, with the access time being the most critical factor given the use of CDMA contention based MAC protocols.

2.3 Related Work

One of the oldest internet protocols is the Network Time Protocol (NTP) [8]. A networking protocol for clock synchronization between computer systems over variable-latency, packet-switched data networks, it is a commonly used mechanism which can achieve global clock synchronization. Since NTP is robust to failures, self-configuring, and also scalable, it has several features which are favourable in wireless sensor networks. However, since it is also server based, as well as energy intensive, these two attributes oppose the design principles of wireless sensor networks such that it is unsuitable for the cause.

2.3.1 Sender-Receiver Synchronization

The traditional clock synchronization approaches employ a scheme known as sender-receiver synchronization. As the name suggests, the synchronization of two nodes is achieved by an exchange of messages between a sender and a receiver; the labels "sender" and "receiver" here refer to the nodes which first send and receive the first message exchange for synchronization. Figure 2-2 shows the most common method of time synchronization between a sender and a receiver using a two message exchange [8].

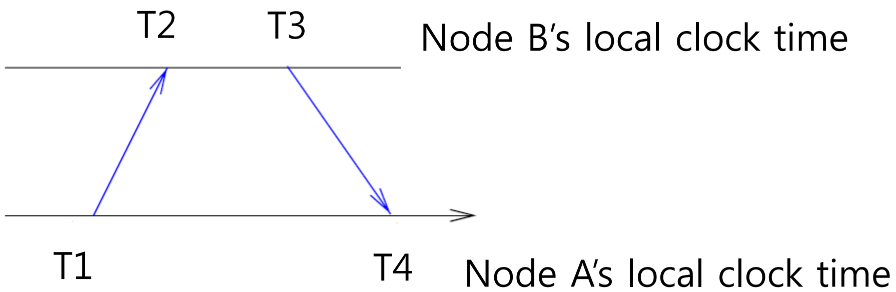

Figure 2-2 - two way message exchange between a pair of nodes

Through the two way message exchange shown in Figure 2-2, node A (the sender) can synchronization its own clock time to that of node B's, such that the two nodes become synchronized (see section 3.1). An important element of this two way message exchange is that the synchronization is initiated by node A (hence this exchange is *sender initiated*), even though A is synchronizing to B. In most networks where a coordinator exists (such as the beacon broadcasting coordinator in TDMA), it is often desired for all slave nodes to synchronize to the coordinator. If this two way message exchange is to be applied directly to a TDMA network, not only would specific time synchronization messages have to be exchanged, but each slave node would also be required to initiate the exchange for time synchronization. The proposed protocol in this thesis uses a modified application of this two way exchange in order to better suit it for TDMA networks (see section 3.1).

The Time-Sync Protocol proposed in [10] uses this two way exchange to synchronize a multi-hop network with tree topology, where the network is divided into multiple levels according to a node's hop level from the coordinator. Firstly level 1 nodes, which are closest to the coordinator, are synchronized to it, and consequently level 2 nodes synchronize to level 1 nodes. In this manner all nodes are synchronized in a flooding manner, level after level.

In [11], the authors also proposed a Tiny-Sync-Mini-Sync protocol based on the two way message exchange noted above, but instead of obtaining an exact estimation for the offset, upon acquiring multiple data points, a lower and upper bound on the relative clock drifts and offsets can be calculated, from which estimates of the true values can be estimated. Although values of the estimates increase as more and more data points are obtained, the drawback of the protocol is its complexity, as there is a need to store many historical data values for each node, as well as having to use certain algorithms to select which data points to store for optimal estimation.

2.3.2 Receiver-Receiver Synchronization

Whereas traditional synchronization protocols are based on sender-receiver synchronization, [12]'s Reference Broadcast Synchronization (RBS) uses a receiver-receiver based synchronization scheme.

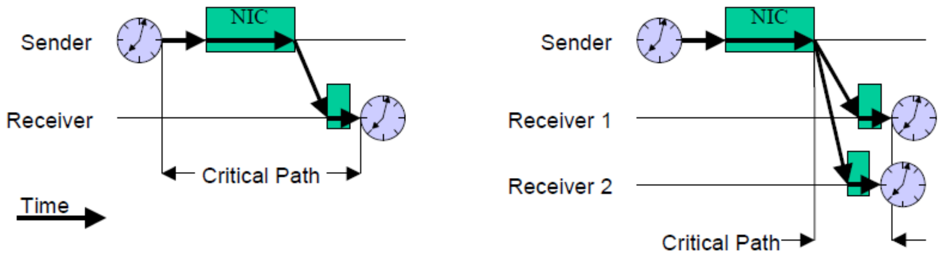

Figure 2-3 - receiver-receiver synchronization showing the shorter critical path compared to sender receiver synchronization

The basic principle in this protocol is the exchange of messages between multiple *receivers* after all receivers have received an external message from a sender. By having a timestamp on the sender's message, receivers use the external message's arrival time as a point of reference for comparing clocks. By removing the sender from the synchronization message exchange, the sender's nondeterminism in terms of send and access time become separated from the critical path of the synchronization process.

2.3.3 Receiver-Only Synchronization

In addition to the sender-receiver and receiver-receiver synchronization schemes mentioned previously, [13] introduces a protocol (Pairwise Broadcast Clock Synchronization) which utilizes both types of synchronization, as well as receiver-only synchronization. A third node which overhears two other nodes exchanging synchronization messages can synchronize to a one of those nodes, albeit with decreased accuracy. The drawbacks of this algorithm is its spatial limitation (all nodes have to be within the Pairwise Synchronization region), and the complexity in the calculation of the joint maximum likelihood estimator of clock offsets and skews. Further work leading from the Pairwise Broadcast Clock Synchronization was carried out in [14], resulting in an application of the protocol in a multi-hop topology, overcoming the spatial limitation of the original scheme, as well as adding some other improvements.

2.3.4 Clock Skew Estimation and Correction

In order to avoid complexity in time synchronization protocols, clock skew estimation is not often considered (such as in [10]). It should be noted, however, that applying a clock skew correction mechanism guarantees the long-term stability of synchronization, allowing for less frequent synchronizations.

A basic method of clock skew estimation is by performing a least-squares linear regression of phase offsets from multiple observations, as used in [12], whereas the technique used in the Pairwise Broadcast protocol [13] uses a maximum likelihood estimator algorithm derived in [15]. In a later section it will be shown that it is possible to easily obtain a reasonable estimation of relative clock skew in TDMA based protocols.

2.3.5 Clock Synchronization in TDMA Based Networks

So far all of the related works mentioned have been for generic sensor network synchronization systems; where there are many different novel ideas for time synchronization, time synchronization especially designed for TDMA MAC based wireless networks have been much less common and limited [16]. One protocol designed for TDMA time synchronization is the Periodic Global Broadcast Time Synchronization (PGB-TS) scheme [17]. Much like the Timing Synchronization Function (TSF) specified in the IEEE 802.11 wireless local area network (WLAN) standard [18], PGB-TS uses beacon frames which contain a timestamp value (the value of the TSF timer in case of 802.11) sent by a coordinator, these beacon frames are received by all slave nodes, which can then adjust their own local clock values according to the timestamp value. PGB-TS improves on TSF by eliminating the most indefinite errors of send time and access time, with the TDMA based MAC properties. By programming the hardware to construct a packet much earlier than its sending time and placing it in a buffer, much of the send and access time errors can be eliminated [17]. PGB-TS also estimates the clock skew error between nodes through a linear regression of the relative offsets between slave nodes and the coordinator decreasing the clock skew error and improving synchronization.

Chapter 3

3. Propagating Chain Time Synchronization for TDMA Based Wireless Sensor Networks

3.1 Overview

Propagating Chain Time Synchronization, PCTS, is a novel synchronization system for TDMA based wireless sensor networks. In PCTS, the main objective is to achieve both lower average errors and worst errors for the offset between slave nodes and the master coordinator node, thus allowing for shorter time slot intervals to minimize wasted bandwidth due to small packet payloads in typical wireless sensor networks. PCTS employs the use of both overhearing and piggybacking in order to achieve chained synchronizations across the network, allowing for the estimation of skew estimation to greater guarantee the long-term stability of synchronization. To further improve the accuracy of the algorithm, skew propagation correction is utilized by taking advantage of the scheme's synchronization sequence, which follows closely with that of the scheduling scheme used in TDMA.

3.2 System Model

3.2.1 Basic Assumptions

Before describing the main features of PCTS, the basic assumptions for the considered wireless sensor network are that of a battery management system (as mentioned in section 1.2), more specifically:

- All sensor nodes are immobile (an in-situ, embedded system).
- All sensor nodes are identical (except from clock frequency differences).
- The radio channel is symmetric.
- The power source of all nodes is not a limiting factor.

3.2.2 Topology

Since a TDMA MAC protocol scheme is used, the topology of the system model can assumed to be a basic star topology; however, the transmitting range of the nodes is flexible; the coordinator should have a transmission and reception range to cover the whole network whilst each sensor node should have a range which covers two other nodes, more specifically the nodes which are assigned time slots before and after the said node's own assigned time slot (this is necessary due to the overhearing of messages as discussed later).

Figure 3-1 shows the required transmission and receiver range of node 2, as it should overhear packets from node 1 and node 3, which send in the time slots before and after node 2's assigned time slot respectively.

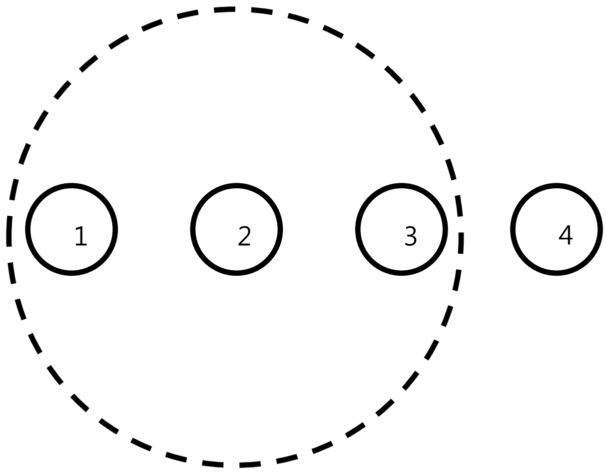

Figure 3-1 - showing the transmission and reception range of a sensor node

3.2.3 Chained Synchronization

The main feature of PCTS is the idea of chaining synchronization throughout the network using two-way message exchanges, between 2 nodes at a time. Consider the two-way message exchange introduced briefly earlier:

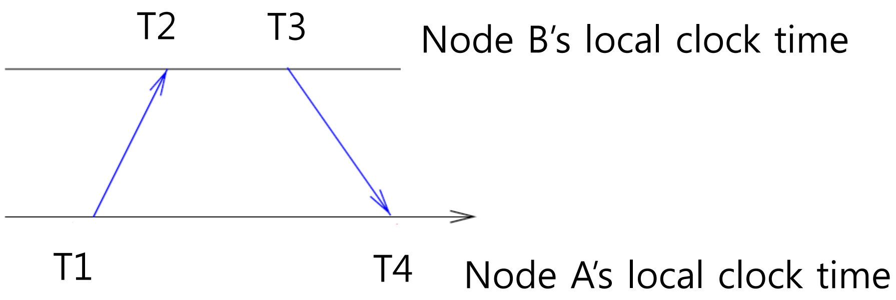

Figure 3-2 - two way message exchange between a pair of nodes

From Figure 3-2 [8] it is possible to obtain the clock offset and propagation delay as:

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2} \qquad (1)$$

and

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \qquad (2)$$

respectively, where the relationship between the two is simply $T2 = T1 + \Delta + d$.

3.2.4 Overhearing and Piggybacking

The benefit of using the two-way exchange method is that only local timestamps need to be stored in order for synchronization to take place. Node B (Figure 3-2) simply stores its local clock value of T2, attaches the timestamp together with T3 in its message back to node A, and then it can disregard all timestamps. Node A, on the other hand, is required to store T1, and upon reception of the reply message from node B, after using all four timestamp values to perform its time offset synchronization to B, can disregard all time stamps. Since time stamping messages on send and receive is common throughout many different protocols [18], it is not an implementation which requires any specific overhead on packets being sent. In this way, the "piggybacking" of timestamps onto data packets is taken advantage of, allowing data packets to be overheard for the PCTS scheme discussed in more detail below.

For simplicity, consider a TDMA based network with 3 sensor slave nodes and one coordinator. Figure 3-3 shows the flow of data between nodes, with a beacon broadcast slot followed by data transfer from assigned nodes in the following slots during each beacon interval. Since a non-contention based TDMA protocol is used, there are no collisions in time slots, and each sensor node is assigned one specific time slot per beacon interval by the coordinator.

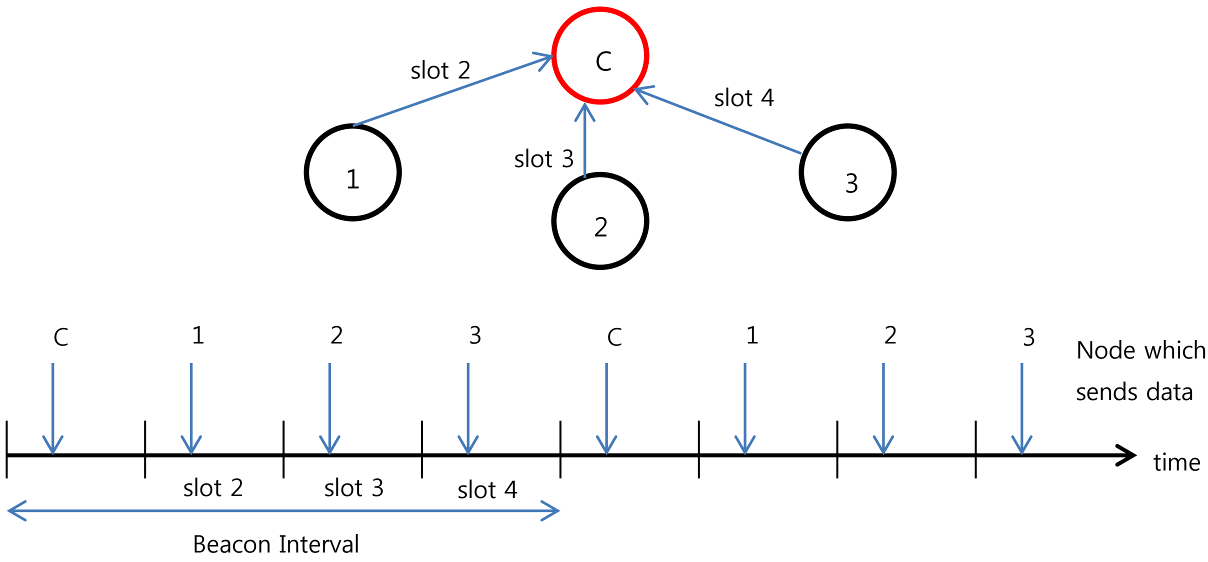

Figure 3-3 - data flow sequence in TDMA

Now suppose node 1 is required to time synchronize to node 2; it is in fact possible to achieve this synchronization once every beacon interval by node 2 overhearing node 1's message in slot 2, and node 1 overhearing the data sent by node 2 to the coordinator in slot 3 (Figure 3-4), recall Figure 3-2. Time stamping described earlier is used in all messages sent, and hence piggybacking is used.

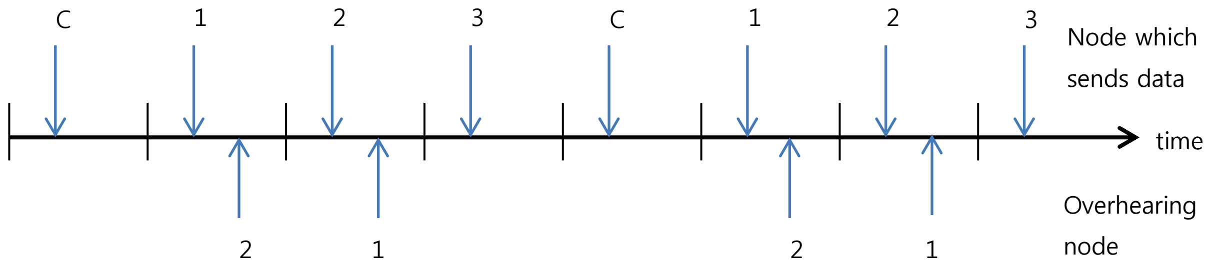

Figure 3-4 - overhearing required for node 2 to synchronize to node 3

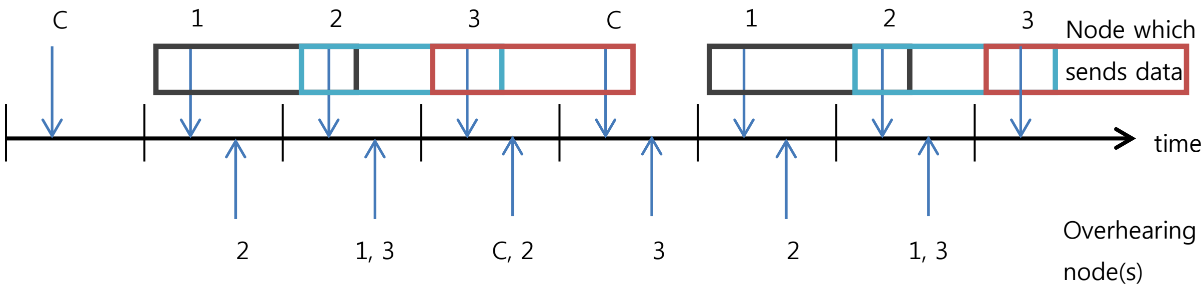

Figure 3-5 - overhearing required for full chained synchronization in network

Figure 3-5 shows that the overhearing of two addition messages is required by each node in order achieve a complete synchronization chain, from the synchronization of node 1 to 2, then 2 to 3, and lastly node 3 to the coordinator, shown in Figure 3-6. It should be noted that the coordinator receives all messages since sensed data is congregated by the coordinator.

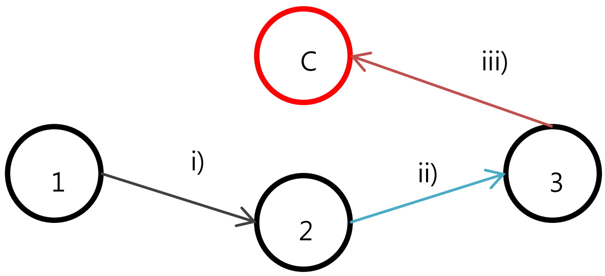

Figure 3-6 - arrows show the order of synchronizations (not data flow),

firstly i) node 1 to 2, then ii) 2 to 3, then iii) 3 to C. Coloured arrows correspond to the coloured boxes shown in Figure 3-5

At the end of each 2 way message exchange, synchronization is achieved by the "sender" initiated node to its adjacent node. Intuitively it can be seen that merely using this chained synchronization technique does not give an improvement over the traditional Time Synchronization Function (TSF) used in 802.11 [802.11]. However, if used in conjunction with TSF, an improvement in the synchronization error can be expected, since the time period between time synchronizations is reduced (hence reducing the amount of time allowed for node clocks to drift apart). In addition, since all nodes synchronize to the coordinator at the beginning of each beacon interval, subsequent chain synchronizations are more accurate because of the updated time values in the sensor nodes.

3.2.5 Propagating Skew Correction

PCTS uses a feature called propagating skew correction in order to achieve lower time synchronization worst errors. By estimating the relative skews between each sensor node and the coordinator, skew correction can guarantee the long-term stability of synchronization, or alternatively require less frequent synchronizations. Estimation of the relative clock skew between adjacent nodes is reasonably simple since the relative offset is calculated using the two-way message exchange method, and the amount of time taken to achieve this offset value is the beacon interval - the time period between a specific node's synchronizations:

$$\text{relative skew} = \frac{\text{relative offset in time period T}}{\text{time period T}} = \frac{\Delta}{BI} \tag{3}$$

Where Δ is the relative offset, and BI is the beacon interval.

Since each node is synchronizing to its neighbouring sensor node, as opposed to the coordinator directly, we should consider an error accumulation problem in chained synchronization. The average clock error will become larger as a node locates farther from coordinator, since it adjusts its clock by using inaccurate time information (similar to a multi-hop synchronization). In order to reduce the accumulated clock skew error, each sensor node will announce its relative skew value to the coordinator, which then broadcasts a propagating skew correction message in its next beacon message, using the algorithm shown in Figure 3-7.

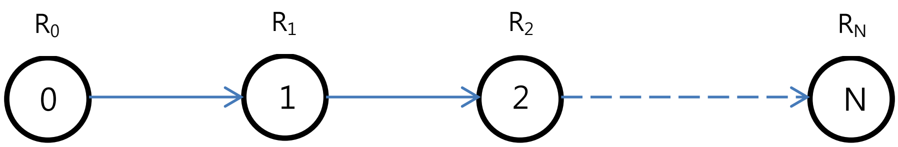

Figure 3-7 - propagating clock skew

By obtaining the relative clock skew between adjacent nodes using equation (3), each node's relative clock with respect to node N can be calculated using the equations:

$$S_{01} = R_0 - R_1 \tag{4}$$

$$S_{12} = R_1 - R_2 \tag{5}$$

$$S_{02} = R_0 - R_2 = S_{01} + S_{12} \tag{6}$$

Hence

$$S_{mN} = \sum_{i=m}^{N-1} S_{i(i+1)} \tag{7}$$

In this manner the relative clock skew of every sensor node can be computed with respect to the coordinator, and broadcasted by the coordinator. In order to achieve clock skew correction, a node simply adjusts its clock value every time slot by multiplying its propagating skew correction value by the time slot duration, and then summing this to its clock value. This skew synchronization can be performed either every beacon interval (in the second slot when no device is synchronizing using the chain method), or when required, depending on the wireless sensor network application and requirements.

Chapter 4

4. Theoretical Error Analysis

4.1 System Models

In this section we attempt to analyse the errors of the different components of PCTS, as well as the analysis of the Time Synchronization Function (TSF). Since most TDMA networks use time synchronization schemes very similar to that of TSF, by comparing the performance errors of PCTS with TSF, a good estimation of its performance advantages can be obtained.

Three error models are considered:

1. *TSF:* A beacon synchronization protocol where sensor nodes synchronize to the coordinator upon receiving the beacon signal sent at the beginning of every beacon interval.

2. *Chained Synchronization:* The scheme described in section 3.2.4, which includes beacon synchronization as well as chained synchronization between adjacent nodes, propagating throughout the network.

3. *Two-way message exchange:* The offset estimation error of the two-way message exchange used in PCTS is analysed. Since both chained synchronization and relative skew

estimation/correction uses the two-way message exchange method, either directly or indirectly (by using the offset or estimating the skew from the offset), the error from the offset estimation affects both synchronization techniques.

4.2 Node Clock Modelling

Software clocks in senor nodes are governed by quartz-governed oscillators. The production of these quartz crystals cannot be all identical, resulting in clock rate (or frequency) discrepancies between nodes. In reality, a node's clock rate may also vary depending on other factors such as temperature, humidity, and supply voltage, age of the quartz, etc., which results in different clock drift rates. Equation (8) shows an example of how temperature can affect clock drift rates:

$$f = f_0 \times \{1 - 0.04 \times 10^{-6} \times (T - T_0)^2\} \tag{8}$$

Where f, f_0, T, and T_0 represent the current frequency (rate), base frequency, temperature and base temperature respectively.

For the purposes of this thesis it is assumed that the frequency, or rate, of all nodes do not change with time; however, even if the frequency rates vary with time, since skew estimation is based on the most recently calculated relative offset values, the accuracy of PCTS is not compromised.

Let us assume the clock frequency of a node to be a normally distributed random variable X with mean μ and variance σ :

$$X \sim N(\mu, \sigma^2)$$

The maximum drift rate ρ is a value given by the manufacturers of quartz (in parts per million, ppm), guaranteeing:

$$1-\rho \leq \text{drift rate} \leq 1+\rho$$

A node with a perfect clock has a drift rate of 1, giving a mean value of $\mu = 1$ and by using the empirical rule, a variance of $\sigma = \frac{\rho}{3}$ guarantees 99.7% of samples to be within the manufacturing tolerance.

Hence we can define the required variables for error analysis:

$X_i \sim N(1, \left(\frac{\rho}{3}\right)^2)$, the frequency rate of any node i , which is normally distributed .

ρ , the maximum drift rate in ppm (specified by quartz manufacturers).

s , the duration of one time slot, during which one assigned node can send data.

n , the number of nodes in the network, which is also the total number of slots in each beacon interval, assuming no wasted slots.

$BI = n \times s$, the duration of each beacon interval, with no wasted time slots.

The following section will attempt to calculate the mean square errors (MSEs) and worst errors for the three system models listed in section 4.1.

4.3 TSF

By considering the local clock value at each time slot, the offsets (relative to the coordinator) from the beginning of the beacon interval till each time slot are, for each node:

$$s(\overline{X_i} - X_i),\ 2s(\overline{X_i} - X_i),\ 3s(\overline{X_i} - X_i),\ \ldots ns(\overline{X_i} - X_i)$$

The mean value $\overline{X_i}$ can be considered as the true clock of the coordinator, and $\overline{X_i} - X_i$ is the relative skew value between node i and the coordinator.

Since time synchronization is performed at the beginning of each beacon interval, the mean square error (MSE) of the offset can be calculated as the sum of the offsets at each time slot over all the nodes in the network.

Summing over all slots in the beacon interval in time, the total offset is:

$$s(\overline{X_i} - X_i) + 2s(\overline{X_i} - X_i) + 3s(\overline{X_i} - X_i) + ...ns(\overline{X_i} - X_i) = \sum_{j=1}^{n} js(\overline{X_i} - X_i)$$

(9)

And summing for every node in the network gives:

$$\sum_{j=1}^{n}\sum_{i=1}^{n} js(\overline{X_i} - X_i) \tag{10}$$

Hence the mean square error (MSE) for TSF is given by:

$$\frac{1}{n}\sum_{j=1}^{n}\frac{1}{(n-1)}\sum_{i=1}^{n}\left(js\left(\overline{X_i} - X_i\right)\right)^2 \tag{11}$$

The first summation is the sum of the offset error in time, for every time slot of the beacon interval, whilst the second summation represents the sum of the relative offsets between each node and the coordinator. Since the coordinator is the point of reference for all other sensor nodes, there exists $n-1$ relative clock offset values between sensor nodes and the coordinator.

The worst error offset value is simply the offset of any given sensor node after the time of $n-1$ slots, before the synchronization phase at the beginning of the beacon interval:

$$\max\left\{(n-1)\,s\left(\overline{X_i} - X_i\right)\right\} \tag{12}$$

4.4 Chained Synchronization

Whereas the time between consecutive synchronizations is the same for all nodes in TSF, the time between synchronization differs for every node in the chained synchronization model. By synchronizing using both the broadcast beacon and chained methods, each sensor node updates its clock value twice per beacon interval. Assuming that the propagation delay of messages received and the time needed to update clock values are negligible, the time drift offset at each time slot are:

Node 2: $s\left(\overline{X_i} - X_2\right)$, $2s\left(\overline{X_i} - X_2\right)$, $s\left(\overline{X_i} - X_2\right)$, $2s\left(\overline{X_i} - X_2\right)$, $3s\left(\overline{X_i} - X_2\right)$, $\ldots (n-2)s\left(\overline{X_i} - X_2\right)$

Node 3: $s\left(\overline{X_i} - X_3\right)$, $2s\left(\overline{X_i} - X_3\right)$, $3s\left(\overline{X_i} - X_3\right)$, $s\left(\overline{X_i} - X_3\right)$, $2s\left(\overline{X_i} - X_3\right)$, $\ldots (n-3)s\left(\overline{X_i} - X_3\right)$

Since the second synchronization for each sensor node (node 1 is the coordinator) - the chain synchronization, takes place in the i^{th} slot for the i^{th} node (figure 10b), the offset distribution during each slot is different for every node, as given by:

Node i: $s\left(\overline{X_i} - X_i\right)$, $2s\left(\overline{X_i} - X_i\right)$, $3s\left(\overline{X_i} - X_i\right)$ $\ldots is\left(\overline{X_i} - X_i\right)$, $s\left(\overline{X_i} - X_i\right)$, $2s\left(\overline{X_i} - X_i\right)$, $3s\left(\overline{X_i} - X_i\right)$, $\ldots (n-i)s\left(\overline{X_i} - X_i\right)$

For $2 \le i \le n-1$, since node 1 is the coordinator, and node n's distribution is simply:

Node n: $s\left(\overline{X_i}-X_n\right)$, $2s\left(\overline{X_i}-X_n\right)$, $3s\left(\overline{X_i}-X_n\right)$, ...
$(n-1)s\left(\overline{X_i}-X_n\right)$

And so summing the clock offsets at each time slot for each node gives:

Node i: $\sum_{j=1}^{i} js\left(\overline{X_i}-X_i\right)+\sum_{j=1}^{n-i} js\left(\overline{X_i}-X_i\right)$, for $2\le i\le n-1$

Node n: $\sum_{j=1}^{n-1} js\left(\overline{X_i}-X_n\right)$

The mean square error (MSE) for chained synchronization can then be estimated as:

$$\frac{1}{n-1}\left[\frac{1}{n}\sum_{i=2}^{n-1}\left\{\sum_{j=1}^{i}\left(js\left(\overline{X_i}-X_i\right)\right)^2+\sum_{k=1}^{n-i}\left(ks\left(\overline{X_i}-X_i\right)\right)^2\right\}+\sum_{l=1}^{n-1}\left(ls\left(\overline{X_i}-X_n\right)\right)^2\right] \tag{13}$$

The worst error offset value for chained synchronization is in fact similar to that of TSF, given by node n in the network:

$$(n-1)s\left(\overline{X_i}-X_n\right) \tag{14}$$

4.5 Two-Way Message Exchange Synchronization Error

Calculating the clock offset values for PCTS is challenging, as perfect clock skew correction would theoretically eliminate any offset error. Instead, it is possible to estimate the error of the skew estimation, since the estimation is based on an estimation of the offset, which is calculate from the two-way message exchange.

Considering the two way message exchange model (Figure 3-2), together with the synchronization errors listed earlier in section 2.2.1, the following expressions can be obtained:

$$T2 = T1 + S_A + P_{AB} + R_B \tag{15}$$

$$t2 = t1 + S_A + P_{AB} + R_B + O_{t1}^{AB} \tag{16}$$

Here uppercase T's refer to the local time measured by nodes A and B in Figure 3-2 respectively, whereas the lowercase t's refer to the real time (measured by an ideal clock); for example, $t1$ is the real time equivalent of $T1$, which is the clock value measured by node B. S, P and R represent the total send, propagation, and receive times respectively; subscripts refer to the specific node's time error delays. The term O_{t1}^{AB} in the second equation represents the clock offset between node A and B at the real time $t1$, since all real time values are used in the second equation.

Similarly:

$$t4 = t3 + S_B + P_{BA} + R_A - O_{t4}^{AB} \tag{17}$$

Noting that $O_{t3}^{BA} \approx O_{t4}^{BA} = -O_{t4}^{AB}$ (assuming the offset change to be negligible between $t3$ and $t4$), we can also see that O_{t1}^{AB} can be broken down into:

$$O_{t1}^{AB} = O_{t4}^{AB} + \lambda_{t1t4}^{AB} \tag{18}$$

where λ_{t1t4}^{AB} is the change in the relative offset between nodes A and B during the time period from $t1$ until $t4$, or in the time period $t4 - t1$.

Through manipulation of the equations above and equation (1), we can obtain:

$$2 \times \Delta = S^{UC} + R^{UC} + P^{UC} + \lambda_{t1t4}^{AB} + (2 \times O_{t4}^{AB}) \tag{19}$$

where $S^{UC} = S_A - S_B$, $R^{UC} = R_B - R_A$, and $P^{UC} = P_{AB} - P_{BA}$ represent the uncertainty at the sender, receiver, and in the propagation time respectively. Since the true value of the offset is given by O_{t4}^{AB} (the value used by node A to synchronize its clock), the error is given by:

$$\text{Error} = \Delta - O_{t4}^{AB} = \frac{S^{UC}}{2} + \frac{P^{UC}}{2} + \frac{R^{UC}}{2} + \frac{\lambda_{t1t4}^{AB}}{2} \tag{20}$$

In our assumption of identical sensor nodes, however, the uncertainty terms would in fact become zero, suggesting that the two way message exchange eliminated uncertainty from the sender, receiver and the propagation time, leaving the error as given by:

$$\text{Error} = \frac{\lambda_{t1t4}^{AB}}{2} \qquad (21)$$

Since the term λ_{t1t4}^{AB} is the change in the relative offset between nodes A and B during the time period $t4-t1$, the term can be further broken down into:

$$\lambda_{t1t4}^{AB} = (t4-t1) \times \text{relative skew between A and B} \qquad (22)$$

By substituting for the relative skew between adjacent nodes, and using the approximation of $t4-t1=s$, we can obtain the two-way message exchange error of:

$$\text{Error} = \frac{s\left(X_A - X_B\right)}{2} \qquad (23)$$

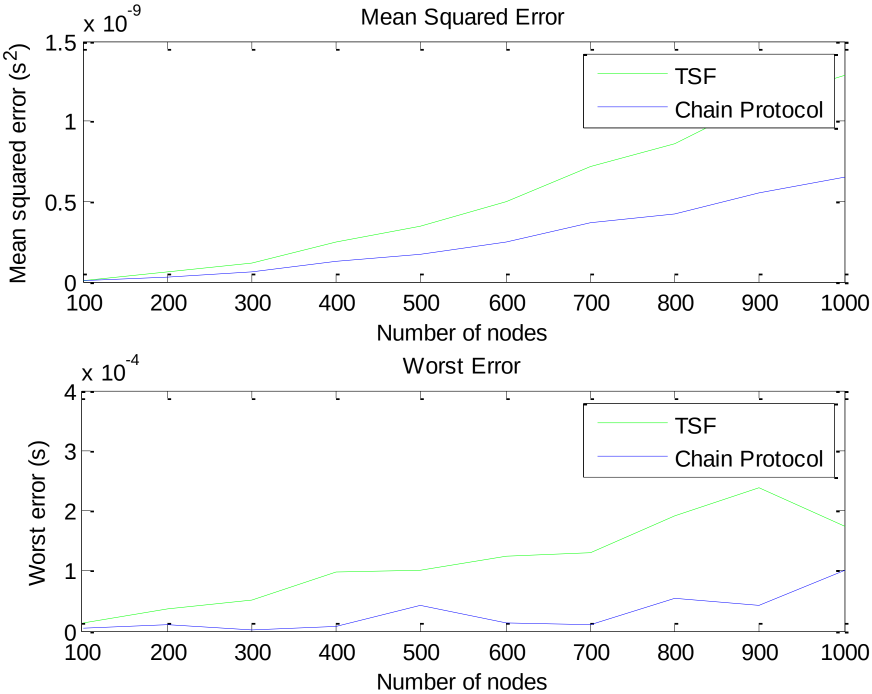

Figure 4-1 - expected errors calculated from theoretical analyses

Chapter 5

5. Simulation

In this section the performance of PCTS is compared to that of TSF and the case of chained synchronization (PCTS without skew estimation). Firstly the real parameters used for simulation are introduced and justified.

5.1 Simulation Parameters

For a realistic evaluation of the different schemes discussed in this thesis, suitable simulation parameters for TDMA networks should be chosen and used. The main standard on which simulation parameters are based on is IEEE 802.15.4 [19].

The IEEE 802.15.4 standard is one designed for wireless personal area networks (WPANs). Often compared to Bluetooth, 15.4's greater support for multiple devices (up to 254 nodes) and range gives it greater flexibility for use in WSNs. The standard specifies the physical and media access control layers, on top of which other standards can be built, such as ZigBee and MiWi.

In the IEEE 802.15.4 MAC protocol, two operational modes can be selected: a non beacon-enabled mode, where non-slotted CSMA/CA is used, and a beacon-enabled mode, where beacons are sent and received periodically throughout the network. The operational mode is selected by a central node in the network defined as a PAN coordinator, and it is this device which broadcasts beacons whilst in beacon-enabled mode. Since the beacon-enabled mode can provide a guaranteed delivery service, this mode operates essentially as a TDMA protocol, albeit *limited to seven devices and 16 time slots.*

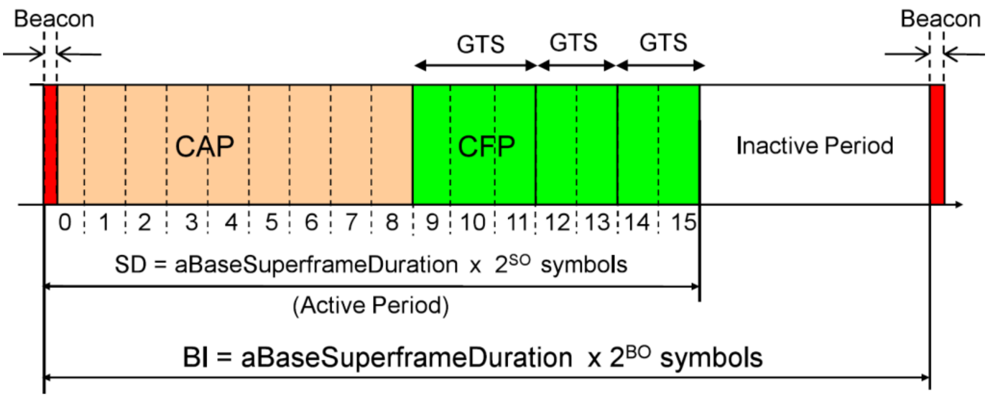

Figure 5-1 - IEEE 802.15.4 superframe structure

The beacon interval (BI) defines the time period between two consecutive beacon frames, and the beacon interval divided into an active and inactive period. The optional inactive period allows for all devices in the network to sleep, thus allowing for energy efficiency. The active period is further divided into a contention access period (CAP) and a contention free period (CFP). Figure 5-1 shows the structure of the superframe, as well as the active period further divided into 16 equal sized timeslots (0 to 15). The length of the beacon interval (BI) and superframe duration (SD) are defined as:

$$BI = aBaseSuperframeDuration \text{ x } 2^{BO} \tag{24}$$

where $0 \leq BO \leq 14$, and

$$SD = aBaseSuperframeDuration \text{ x } 2^{SO} \tag{25}$$

where $0 \leq SO \leq BO \leq 14$.

The minimum length of the superframe is defined in the standard by aBaseSuperframeDuration, which is fixed at 960 symbols.

$$\text{Duty Cycle} = \frac{SD}{BI} \tag{26}$$

The duty cycle is defined as the ratio between the superframe duration and the beacon interval, indicating the amount of time for which the beacon interval is not used.

In equations (24) and (25) it can be seen that the parameters BO and SO (beacon order and superframe order respectively) can be chosen to decrease and increase the length of the beacon interval dramatically. However, no matter what the value of SO chosen, the number of timeslots remains fixed at 16.

The beacon-enabled mode with a duty cycle of one is essentially a TDMA protocol limited to 16 time slots (which can vary in time length) and seven devices.

BO=SO	BI=SD (sym)	BI=SD (s)	1 slot duration
0	960	0.01536	0.00096
1	1920	0.03072	0.00192
2	3840	0.06144	0.00384
3	7680	0.12288	0.00768
4	15360	0.24576	0.01536
5	30720	0.49152	0.03072
6	61440	0.98304	0.06144
7	122880	1.96608	0.12288
8	245760	3.93216	0.24576
9	491520	7.86432	0.49152
10	983040	15.72864	0.98304
11	1966080	31.45728	1.96608
12	3932160	62.91456	3.93216
13	7864320	125.82912	7.86432
14	15728640	251.65824	15.72864

Table 3 - showing standard supported time slot durations

To simulate TCF and PCTS, the simulation parameter of the time slot duration are based on the standard supported values shown in table 3, since they are known to be supported by hardware devices such kmotes and telosb motes. However, whereas 15.4 limits the number of devices to seven, and the number of time slots to 16, our simulation will be flexible in the number of time slots and devices. Since each device is assigned one time slot to send data, the number of devices equal the number of time slots in each beacon interval.

The required time slot length is flexible, and depends on the amount of data required to be sent during one time slot. For a typical packet size of 48 bytes (the size supported in the telosb specification sheet), assuming a 250kbps data rate in the 2.4GHz frequency band as given in the physical layer 15.4 specification, a required timeslot length size is typically 1.92ms.

For the simulation of the sensor node clocks, the maximum drift rate ρ is taken as 100ppm, an average value given by quartz manufacturers.

5.2 Simulation Results

Simulated as a TDMA MAC protocol network in Matlab, the mean average error and worst error of TSF, chained synchronization and PCTS were compared.

Figure 5-2 shows the simulation results of the three system models:

- TSF – the protocol which employs only beacon synchronization at the beginning of every beacon interval.
- Chain protocol – in addition to the beacon synchronization utilized in TSF, in the chain protocol adjacent nodes also synchronize to each other through the chain method described in section 3.2.4.

- PCTS – the proposed protocol utilizes beacon synchronization, and carries out the chain two-way message exchanges between adjacent nodes, but instead of updating sensor node's offsets with the two-way exchange method, a relative skew estimation and correction scheme is used.

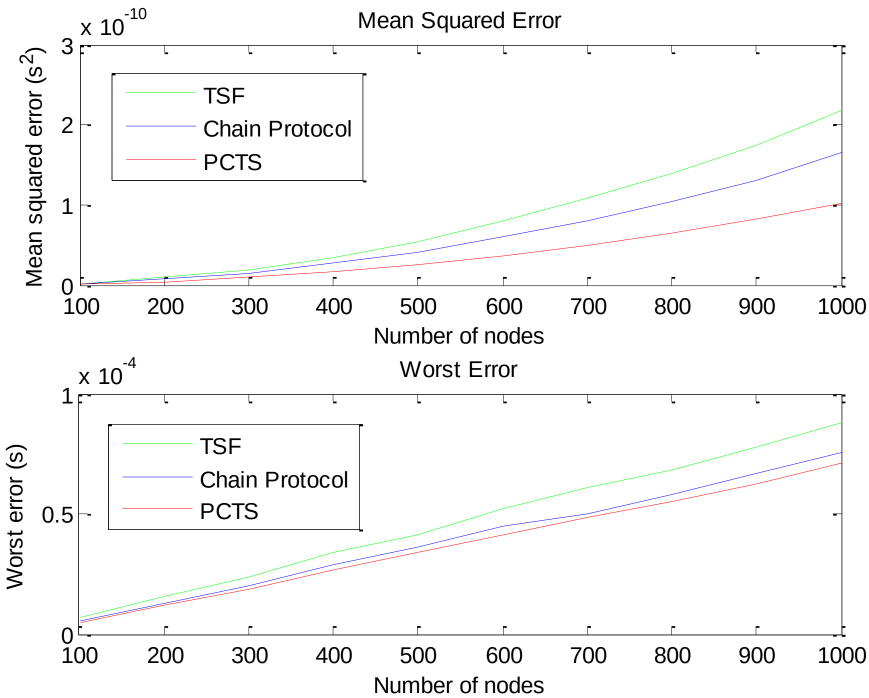

Figure 5-2 - errors with changing number of nodes, for a fixed time slot duration

A time slot value of $s = 1.92$ ms is used for the simulation in Figure 5-2, over networks ranging from 100 nodes to 1000 nodes. The performance of TSF accurately follows that of the theoretical analysis in section 4.3, for both the mean squared error and the worst error. As expected, the performance of the chain protocol improves upon TSF, but the improvement is slighter than in the theoretical prediction; the reasoning for this is likely due to the reduced accuracy of the second time synchronization (chain synchronization) for every node in each beacon interval. Since chain synchronization synchronizes adjacent nodes, the node which is node "B" in the two-way message exchange (see Figure 3-2) is the reference for the synchronization; however, node B's clock would also have a drift error relative to the coordinator node, since an amount of time would have passed (since receiving the beacon message) depending on node B's assigned slot time in the beacon interval.

PCTS, which employs skew estimation and correction, provides further error improvements; however, there is only a slight improvement in the worst error over the chain synchronization scheme. For both proposed schemes of chain synchronization and PCTS, the improvement in the synchronization errors (compared to TSF) increases as the number of nodes increases.

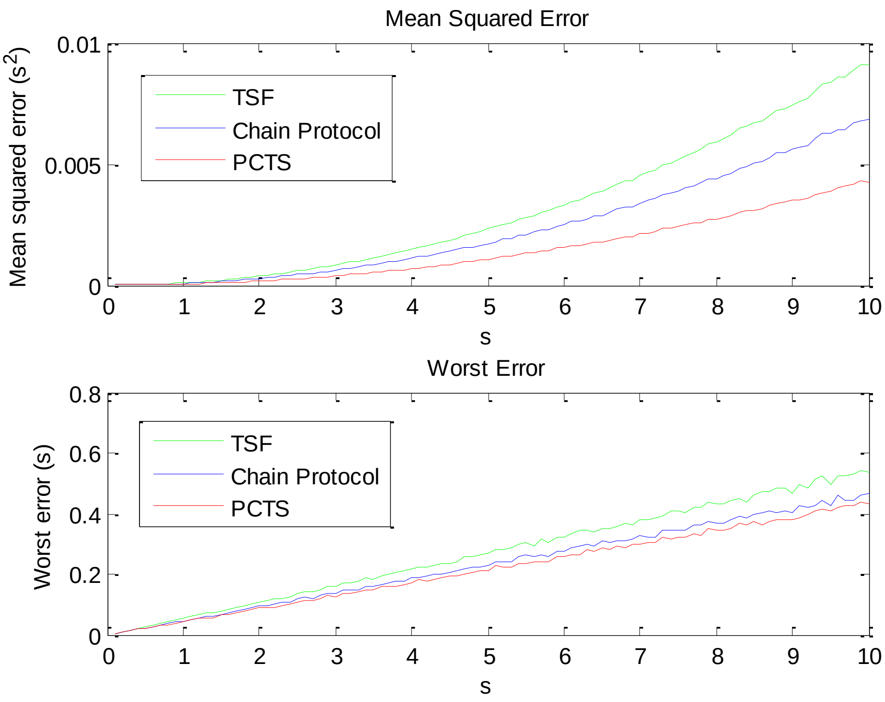

Figure 5-3 - errors with varying time slot lengths for a fixed number of nodes

Figure 5-3 compares the error of the three protocol schemes for a network of $n = 500$ with varying time slot durations, up to 1 second. It is seen that the improvement in the errors is similar to the case of varying the number of nodes in the network.

The conclusion which can be drawn from Figure 5-2 and Figure 5-3 is that the proposed schemes of the chain protocol and PCTS provide an improvement in the synchronization error compared to traditional TDMA beacon synchronization schemes such as TSF.

This improvement is further enhanced as the number of nodes and the time slot duration are both increased.

Since TDMA networks have no collisions, the latency of the network is determined by the beacon interval, which is in turn decided by the design specifications required for the sensor network application. The required time slot length is determined from the packet size required to be sent, whereas the number of devices determines the number of time slots in a beacon interval. For a given time slot length, the proposed schemes are able to support more devices than TSF due to lower synchronization errors, whereas for a given number of devices, the proposed protocols are able to support longer slot time lengths.

Packet Overhead

Since the PCTS scheme utilizes skew estimation and correction, the coordinator is required to broadcast the skew propagation correction value for each node in its beacon broadcast signal; in addition, sensor nodes are required to broadcast their relative offset values with the adjacent node to the coordinator. Implementing skew correction, although improving the synchronization error, will result in extra pack overheads resulting in wasted bandwidth resources (or even lengthening the time slot duration in order to accommodate the extra overhead). Since the chained synchronization scheme only uses piggybacking and overhearing, without extra broadcast information necessary from the coordinator, no extra overheads are necessary, whilst improving accuracy.

Chapter 6

6. Conclusion

This thesis has introduced the concept of different types of wireless sensor networks as well as their various applications. For more specific types of in-situ networks, such as the battery management system discussed, periodic, reliable data sensing and transmission is required; for such a system, a non-contention based MAC layer protocol is suitable, as it avoids packet collisions, as well as allows for appropriate scalability.

One suitable MAC layer protocol is time division multiple access (TDMA), where a channels is divided into timeslots, each slot assigned to a node device in order to send data to the coordinator. In this case, the time synchronization of nodes is important in order to avoid collisions resulting from each individual node's clock drifts. The traditional method of time synchronization in TDMA networks is one similar to the time synchronization function in the IEEE 802.11 standard, essentially a beacon broadcast based synchronization scheme. In this thesis, Propagating Chain Time Synchronization (PCTS) was proposed, a novel scheme which employs the use of a propagating two-way message exchange between adjacent nodes, in order to estimate the relative skew value between each sensor node and the coordinator. Skew estimation is possible through a skew propagation scheme, and the use of piggybacking timestamps, as well as the overhearing of messages, saves resources and bandwidth when applying PCTS. It was also seen that PCTS provides an improvement in the synchronization error compared to traditional TDMA beacon synchronization schemes such as TSF. This improvement is further enhanced as the number of nodes and the time slot duration are both increased, meaning that for a given time slot length, the new scheme is able to support more devices due to lower synchronization errors, and for a given number of devices, the proposed protocols are able to support longer slot time lengths.

Bibliography

[1] IF Akyildiz, W Su, Y Sankarasubramaniam, E Cayirci, "A survey on sensor networks," in *Communications magazine, IEEE 40 (8)*, 2002.

[2] C. C. Enz, A. El-Hoiydi, J-D Decotignie, V. Peiris, "WiseNET: An Ultralow-Power Wireless Sensor Network Solution," *IEEE Computer,* vol. 37, no. 8, pp. 685 - 692, 2004.

[3] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/AMA Transactions on Networking,* vol. 12, no. 3, pp. 493 - 506, 2004.

[4] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy-efficient, Collision-Free Medium Access Control for Wireless Sensor networks," *Proc. ACM SenSys 03,* pp. 181 - 192, 2003.

[5] K. Jamieson, H. Balakrishnana, and Y. C. Tay, "Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks," *MIT Laboratory for Computer Science, Tech. Rep. 894,* 2003.

[6] G. Lu, B. Krishnamachari, C.S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," *Proceedings of 18th International Parallel and Distributed Processing Symposium,* p. 224, 2004.

[7] I. Demirkol, C. Ersoy, F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine,* vol. 44, no. 4, pp. 115 - 121, 2006.

[8] S. PalChaudhuri, A. Saha, D.B. Johnson, "Probabilistic clock synchronization service in sensor networks," *IEEE Transactions on Networking,* vol. 2, no. 2, pp. 177 - 189, 2003.

[9] D.L. Mills, "Internet time synchronization: The network time protocol," *IEEE Transactions on Communications,* vol. 39, no. 10, pp. 1482 - 1493, 1991.

[10] S. Ganeriwal, R. Kumar, M B. Srivastava, "Timing-sync protocol for sensor networks," *First ACM Conference on Embedded Networked Sensor Systems (SenSys),* 2003.

[11] M. Sichitiu, C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," *IEEE Wireless Communications and Networking Conference,* 2003.

[12] J. Elson, L. Girod, D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002),* 2002.

[13] K.L. Noh, E. Serpedin, "Pairwise Broadcast Clock Synchronization for Wireless Sensor Networks," *Proceedings of IEEE International Symposium on World of Wireless, mobile and Multimedia Networks,* pp. 1 - 6, 2007.

[14] K.L. Noh, Q. Chaudhari, E. Serpedin, B. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Transactions on Communications,* vol. 55, no. 4, 2007.

[15] S.H. Lee, L. Choi, "Chaining Clock Synchronization: an Energy-Efficient Clock Synchronization Scheme for Wireless Sensor Networks," *10th International Symposium on Pervasive Systems, Algorithms and Networks,* 2009.

[16] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, "Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks," *ACM MOBICOM,* pp. 272 - 286, 2001.

[17] I. Bekmezci, F. Alagoz, M.G. Arslan, "Periodic Global Broadcast Time Synchronization (PGB-TS) for TDMA Based Sensor Networks," *IEEE 3rd International Conference on Recent Advances in Space Technologies,* pp. 531 - 536, 2007.

[18] "IEEE Std 802.11™-2012".

[19] "IEEE 802.15.4 Standard".

한글초록

TDMA 기반 무선 센서 네트워크를 위한 시간 동기화 프로토콜

서울대학교 대학원
전기컴퓨터공학부
Eric Yip

최근에 무선 센서 네트워크가 그것의 다양한 이용 가능성 때문에 많이 주목 받고 있다. 무선 센서 네트워크 MAC 계층 프로토콜에서의 많은 연구에도 불구하고 이런 연구들은 주로 프로토콜의 파워 감소나 효율성에 집중되어 왔다. 예를 들어, 배터리 조절 시스템과 같은 많은 유연성을 요구하지 않는 센서 네트워크에서는 에너지가

항상 가장 중요한 요소가 아니라, 신뢰성과 확장성이 보다 중요한 요소가 된다. 이처럼 전통적인 TDMA 프로토콜은 좋은 옵션으로서 고려될 수 있다.

무선 센서 네트워크의 시간 동기화는 학문적으로 많이 연구가 되었지만 TDMA 네트워크에서의 시간 동기화와 관련된 연구는 많이 진행되지 않고 있다. 본 논문에서는 TDMA를 기반으로 하는 무선 센서 네트워크를 위한 시간 동기화 프로토콜이 제안된다. 우리는 그것을 Propagating Chain Time synchronization이라 부른다.

Propagating Chain Time synchronization은 TDMA를 기반으로 하는 네트워크를 동기화하는 새로운 프로토콜이다. 이 방법은 piggybacking과 overhearing을 사용하여 연쇄적인 두 경로의 메시지 교환으로부터 측정된 왜곡된 교정을 통한 전통적인 비콘 동기화와 비교하여 더 좋은 성능을 만족한다.

키워드 : 무선 센서 네트워크, 시간 동기화, TDMA

학번 : 2011-24082