NEW ADVANCES IN DESIGNING ENERGY EFFICIENT TIME SYNCHRONIZATION SCHEMES FOR WIRELESS SENSOR NETWORKS

A Dissertation

by

KYOUNG LAE NOH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Major Subject: Electrical Engineering

NEW ADVANCES IN DESIGNING ENERGY EFFICIENT TIME SYNCHRONIZATION SCHEMES FOR WIRELESS SENSOR NETWORKS

A Dissertation

by

KYOUNG LAE NOH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Erchin Serpedin
Committee Members,	Costas N. Georghiades
	Andrew Chan
	Mikyoung Jun
Head of Department,	Costas N. Georghiades

August 2007

Major Subject: Electrical Engineering

ABSTRACT

New Advances in Designing Energy Efficient Time Synchronization Schemes for Wireless Sensor Networks. (August 2007)

Kyoung Lae Noh, B.S., Hanyang University, Seoul, Korea;

M.S., Hanyang University, Seoul, Korea

Chair of Advisory Committee: Dr. Erchin Serpedin

Time synchronization in *wireless sensor networks* (WSNs) is essential and significant for maintaining data consistency, coordination, and performing other fundamental operations, such as power management, security, and localization. Energy efficiency is the main concern in designing time synchronization protocols for WSNs because of the limited and generally nonrechargeable power resources. In this dissertation, the problem of time synchronization is studied in three different aspects to achieve energy efficient time synchronization in WSNs.

First, a family of novel joint clock offset and skew estimators, based on the classical two-way message exchange model, is developed for time synchronization in WSNs. The proposed joint clock offset and skew correction mechanisms significantly increase the period of time synchronization, which is a critical factor in the overall energy consumption required for global network synchronization. Moreover, the *Cramer-Rao* bounds for the maximum likelihood estimators are derived under two different delay assumptions. These analytical metrics serve as good benchmarks for the experimental results thus far reported.

Second, this dissertation proposes a new time synchronization protocol, called the Pairwise Broadcast Synchronization (PBS), which aims at minimizing the number of message transmissions and implicitly the energy consumption necessary for global

synchronization of WSNs. A novel approach for time synchronization is adopted in PBS, where a group of sensor nodes are synchronized by only overhearing the timing messages of a pair of sensor nodes. PBS requires a far smaller number of timing messages than other well-known protocols and incurs no loss in synchronization accuracy. Moreover, for densely deployed WSNs, PBS presents significant energy saving.

Finally, this dissertation introduces a novel adaptive time synchronization protocol, named the Adaptive Multi-hop Timing Synchronization (AMTS). According to the current network status, AMTS optimizes crucial network parameters considering the energy efficiency of time synchronization. AMTS exhibits significant benefits in terms of energy-efficiency, and can be applied to various types of sensor network applications having different requirements.

To my parents and in memory of my grandmother.

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartful gratitude to my advisor, Dr. Erchin Serpedin, for his guidance, understanding, support, and patience during my graduate studies ay Texas A&M. He has been always an excellent teacher in classes and a great advisor in encouraging me to accomplish this research work. I would also like to thank Dr. Costas N. Georghiades, Dr. Andrew Chan, and Dr. Mikyoung Jun for serving on my academic committee.

I will never forget people in the Wireless Communications Laboratory (WCL). I want to show my appreciation to them for all their help and support. Among my colleagues, special thanks go to Yik-Chung, Qasim, Yi, Ilkay, Salim, and Lingjia for valuable discussions and insightful suggestions on my research and study throughout my years at Texas A&M. I also thank Huseyin, Jaehan, Sangwoo, Wentao, Kwadwo, and Parimal for helping me not feel lonely and being sincere friends.

I have to also thank my friends, Yun-Hwan, who was a perfect roommate, Yeo-Jun, who most frequently hung out together with me in my first two years, Jae-Rock, one of my hearty old school seniors, and all former and current teammates of the KASA for refreshing and recharging me when I was in trouble.

I am also greatly indebted to many people in Korea: First, I would like to warmly thank U-IN-ME brothers who initially motivated me to begin this journey and have never stopped encouraging me. I am also thankful to my colleagues and former team members at SK telecom, my class and lab mates at Hanyang University, for their consistent support and welcomes whenever I temporarily stayed in Korea.

Last, but not least, I am very grateful to my parents and family whose love and unconditional support enabled me to complete this work. This dissertation is dedicated in memory of my grandmother who passed away during my study.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivations

1. Wireless Sensor Networks

With the help of recent technological advances in micro-electro-mechanical systems (MEMS) and wireless communications, low-cost, low-power, and multi-functional wireless sensing devices have been developed. When these devices are deployed over a wide geographical region, they can collect information about the environment and efficiently collaborate to process such information by forming a distributed communication network, called the *wireless sensor network* (WSN). WSN is a special case of wireless ad-hoc network, and assumes a multi-hop communication framework with no common infrastructure, where the sensors spontaneously cooperate to deliver information by forwarding packets from a source to a destination. The feasibility of WSNs keeps growing rapidly, and WSNs have been regarded as fundamental infrastructures for future ubiquitous communications due to a variety of promising potential applications: monitoring the health status of humans, animals, plants and environment, control and instrumentation of industrial machines and home appliances, homeland security, detection of chemical and biological threats and leaks, etc. [1]-[4].

When designing sensor networks, there are a number of important factors to be considered such as tolerance to node failures, scalability, dynamic network topology, hardware constraints, production cost and power consumption [1]. In general, the lifetime of a sensor network is proportional to that of battery since the sensor nodes

---

The journal model is *IEEE Transactions on Automatic Control.*

are usually inaccessible after deployment. Besides, due to the space limitations and other practical constraints in sensor nodes, power is a scarce resource for practical WSNs. For these reasons, the energy efficiency represents in general the top priority in designing WSNs among all the above mentioned design considerations. Data communication is one of the most significant operations in WSNs and assumes a huge portion of the overall energy consumption. Indeed, the energy required for data communication is by far greater than the energy required for data processing in a sensor node [4]. This dissertation focuses on the problem of time (clock) synchronization, which is critical and mandatory for data communications, and one of the most important features for developing energy efficient sensor networks as well.

2. Importance of Time Synchronization in WSNs

Time synchronization is a procedure for providing a common notion of time across a distributed system. It is crucial for WSNs in performing a number of fundamental operations, such as

- Data fusion: Data merging is a main operation in all distributed networks for processing and integrating in a meaningful way the collected data, and it requires some or all nodes in the network to share a common time scale.

- Power management: Energy efficiency is a key designing factor for WSNs since sensors are usually left unattended without any maintenance and battery replacement for their lifetimes after deployment. Most energy-saving operations strongly depend on time synchronization. For instance, the duty cycling (sleep and wake-up modes control) helps the nodes to save huge energy resources by spending minimal power during the sleep mode. Thus, network-wide synchronization is essential for efficient duty cycling and its performance is proportional

to the synchronization accuracy.

- Transmission scheduling: Many scheduling protocols require time synchronization. For example, Time Division Multiple Access (TDMA) scheme, one of the most popular communications schemes for distributed networks, is only applicable to a synchronized network.

Moreover, many localization, security and tracking protocols also demand the nodes to timestamp their messages and sensing events. Therefore, time synchronization appears as one of the most important research challenges in the design of energy-efficient WSNs.

In general, synchronization is considered a critical problem for distributed wireless ad-hoc networks due to their decentralized nature and the timing uncertainties introduced by the imperfections in hardware oscillators and message delays in physical- and MAC-layers. All these uncertainties cause the local clocks of different nodes to drift away from each other over the course of a time interval. In the context of Internet (the most popular distributed network), time synchronization has been thoroughly studied and investigated in the literature. For Internet, the Network Time Protocol (NTP) [5] is employed ubiquitously due to its diverse advantages, such as scalability, robustness and self-configurability. The main advantages of NTP are that it does not rely on GPS and is a software-based protocol which makes it flexible to the type of hardware platforms [6]. However, NTP presents a number of challenges when applied to WSNs due to the unique nature of sensor networks: limited power resources, wireless channel conditions, and dynamic topology caused by mobility and failure. Therefore, different types of synchronization schemes have to be explicitly designed for WSN applications to cope with these challenges (see also the surveys in [7]-[13] for additional motivations in this direction). In this dissertation, we study the

problem of time synchronization for WSNs in three distinct areas aiming at achieving energy-efficient global synchronization.

B. Outline and Contributions of This Dissertation

The main contents and contributions of this dissertation are summarized as follows.

In Chapter II, the general clock model for time synchronization is first introduced and analyzed. Some important features that have to be considered when designing time synchronization protocols for WSNs are presented. Besides, various delay components in timing message delivery are categorized. Chapter III presents three general and fundamentally different time synchronization approaches, namely, sender-receiver, receiver-receiver, and receiver-only synchronization. These basic approaches are analyzed and compared to illustrate the common and different characteristics in clock synchronization of WSNs. Chapter IV categorizes and surveys the existing time synchronization protocols for WSNs focusing mainly on the signal processing aspects including the most recent developments in this field, and relates them to the results presented in Chapter III. In addition, Chapter IV presents the results concerning the importance and effectiveness of adaptive time synchronization schemes, and introduces some important adaptive synchronization protocols as well.

In Chapter V, we study the joint clock offset and skew estimation mechanism based on the two-way timing message exchange model assuming two different classes of random delays in message delivery. A thorough analysis of the classical two-way message exchange model between two nodes under the symmetric exponential and normal noise models is carried out in this chapter. The contribution of this study is threefold. First, we analyze and derive the maximum likelihood estimators (MLEs) and corresponding Cramer-Rao lower bounds (CRBs) for the conventional clock offset

model as described in [14], assuming Gaussian and exponential models for the random delays, respectively. Second, we derive the joint maximum MLE and corresponding CRB using a more realistic linear clock skew model assuming Gaussian random delays. Third, novel and practical clock skew estimators, which do not require to know the fixed portion of delays, are proposed. The introduction of a clock skew correction mechanism prolongs the re-synchronization period significantly, and therefore far less power resources will be required in the synchronization process. In fact, the proposed clock synchronization mechanism can be directly applied to the conventional protocols using simple and low complexity modifications, a feature which is very attractive for WSNs consisting of cheap and small nodes.

Chapter VI proposes a novel time synchronization scheme referred to as the Pairwise Broadcast Synchronization (PBS) protocol, which efficiently combines the sender-receiver synchronization and receiver-only synchronization (newly developed in this dissertation) approaches to achieve network-wide synchronization with a significantly reduced number of synchronization messages, i.e., with lesser energy consumption. This study brings two main contributions: 1. Development of a novel synchronization approach which can be partially or fully applied for implementation of new synchronization protocols and for improving the performance of existing time synchronization protocols. 2. Design of a time synchronization scheme which significantly reduces the overall network-wide energy consumption without incurring any loss of synchronization accuracy compared to other well-known schemes.

The extension of PBS to general multi-cluster sensor networks is also studied in this chapter. Based on a hierarchical connection tree of the network, we propose an energy-efficient pair selection algorithm which only investigates the connectivity among children nodes in every parent-children group of the spanning tree. This chapter shows that the proposed algorithm does not require a heuristic network connection

search and can be easily combined with other level-based protocols by simply adding a group-wise connection discovery procedure.

In Chapter VII, we propose an energy-efficient Adaptive Multi-hop Timing Synchronization (AMTS) scheme with the goal of achieving a long-term network-wide synchronization with minimal energy consumption. The main advantages of the proposed AMTS scheme are as follows. First, it represents a significantly enhanced extension of the popular Timing-sync Protocol for Sensor Networks (TPSN) protocol [14] aiming at minimizing the overall energy consumption in large-scale and long-lived sensor networks. Second, it is equipped with flexible mechanisms to adjust the synchronization mode, the period of network-wide timing synchronization (re-synchronization rate), and joint clock offset and skew estimators in order to achieve long-term reliability of synchronization. Finally, as opposed to some other well known protocols that perform very poorly in high-latency acoustic networks, AMTS provides excellent performance for networks characterized by high propagation delays and possible clock skew variations, e.g., underwater acoustic sensor networks. In addition, the proposed synchronization scheme helps to reduce significantly the energy consumption compared to the conventional protocols and to increase the re-synchronization period, which induces highly energy-efficient timing synchronization. Moreover, the adaptive features present in AMTS make it applicable to various other types of wireless sensor networks with different requirements and design objectives.

CHAPTER II

SIGNAL MODELS FOR TIME SYNCHRONIZATION

A. Definition of Clock

Every individual sensor in a network has its own clock. The counter in a sensor is increased in accordance with the zero-crossings or the edges of the periodic output signal of the local oscillator. When the counter reaches a certain threshold value, an interrupt is created and delivered to the memory. The frequency of the oscillator and the threshold value determine the resolution of the clock. Ideally, the clock of a sensor node should be configured such that $C(t) = t$, where t stands for the ideal or reference time. However, due to the imperfections of the clock oscillator, the clock function of the i^{th} node is modeled as

$$C_i(t) = \theta_{\mathrm{o}} + \theta_{\mathrm{s}} \cdot t + \epsilon, \tag{2.1}$$

where the parameters θ_{o} and θ_{s} are called clock offset (phase difference) and clock skew (frequency difference), respectively, and ϵ stands for random noise.

Assuming the effect of random noise ϵ is negligible, from (2.1), the clock relationship between two nodes, say *Node 1* and *Node 2*, can be represented by

$$C_1(t) = \theta_{\mathrm{o}}^{(12)} + \theta_{\mathrm{s}}^{(12)} \cdot C_2(t),$$

where $\theta_{\mathrm{o}}^{(12)}$ and $\theta_{\mathrm{s}}^{(12)}$ are the relative clock offset and skew between *Node 1* and *Node 2*, respectively. Thus, $\theta_{\mathrm{o}}^{(12)} = 0$ and $\theta_{\mathrm{s}}^{(12)} = 1$ when the two clocks are perfectly synchronized. Suppose there are L nodes in the network, then the global network-wide synchronization is achieved when $C_i(t) = C_j(t)$ for all $i, j = 1, \cdots, L$.

Time synchronization in wireless sensor networks is a complicated problem due

to the following reasons. First, every single oscillator has its unique clock parameters regardless of its type. For instance, according to the data-sheet of a typical crystal-quartz oscillator commonly used in sensor networks, the frequency of a clock varies up to 40 *ppm*, which means clocks of different nodes can loose as much as 40 *ms* in a second. In other words, every single oscillator might assume a different skew parameter ranging from -20 to 20 *ppm*.

Notice that in general, the clock skew θ_s is a time-dependent random variable (RV) and there are two concepts used often in clock terminology regarding the nature of time-dependent randomness present in clock parameters. These concepts are referred to as *short-term* and *long-term* stabilities, respectively. For the oscillators currently used in sensor networks, all these parameters are almost constant for short-term time intervals [15]. Besides, the total power of the noise process is too small to be effective in short time-spans [16]. Therefore, the parameters of a clock are assumed to be constant for the time period of interest.

As far as the long-term stability is concerned, the clock parameters are subject to changes due to environmental or other external effects such as temperature, atmospheric pressure, voltage changes, and hardware aging [15]. Hence, in general, the relative clock offset keeps changing with time, which means that the network has to perform periodic time re-synchronization to adjust the clock parameters.

B. Design Considerations

Time synchronization for conventional wired networks has been thoroughly studied and a plethora of synchronization protocols have been proposed as surveyed in [1]. For wireless sensor networks, there are a number of unique and important factors to be considered when designing time synchronization protocols as listed below.

- *Energy consumption*

  Energy consumption is momentous in wireless sensor networks due to their limited and generally non-rechargeable power resources [17]. Hence, the design of wireless sensor networks should be subjected to maintaining minimal energy expenditure in each sensor node. Various types of power control procedures, such as sleep/wake-up modes and dynamic routing controls, are commonly considered in this regard. Time synchronization is one of the critical components contributing to energy consumption due to the highly energy consuming radio transmissions for achieving clock synchronization. Indeed, the energy consumption required for time synchronization of a node is approximately 17 % of the total energy spent by a node [18]. *Pottie* and *Kaiser* showed in [4] that the RF energy required to transmit 1 bit over 100 meters (i.e., 3 Joules) is equivalent to the energy required to execute 3 millions of instructions. Therefore, developing efficient synchronization algorithms represents an ideal mechanism for trading computational energy for reduced (RF) communication energy. In sequel, energy efficiency is the main concern in designing time synchronization protocols.

- *Latency*

  Latency in message delivery is a fundamental factor when designing communications networks. For the networks based on multi-hop transmissions like wireless sensor networks, this is even more critical because the uncertainty in message delivery significantly increases as the number of hops increases. Besides, the effects of channel variations, mobility, and ad-hoc nature of wireless sensor networks make this problem more complex. Efficient localization and time synchronization protocols are necessary for reducing the latency error and

jitter.

- *Security and reliability*

  Network security has gained a huge attention in recent years as the networks become more accessible and vulnerable due to the development of sophisticated spying techniques and devices. Besides, unlike wired networks, far more frequent message losses occur in wireless networks because of the time-varying nature of wireless channels. Therefore, a mechanism to cope with message losses and malicious attacks in time synchronization will be necessary for wireless sensor networks.

- *Network topology changes*

  The performance of a time synchronization protocol is closely related to the network topology, i.e., it varies with the density and distribution of sensors in the network. Therefore, any shift in the location or scale of sensors incurs a network topology change, which requires at its turn a new self-configuration. Mobility of the sensors and battery timeouts are the main reasons for this change. Hence, for dynamic sensor networks, time synchronization protocols should be able to adapt well to frequent network topology changes.

- *Scalability*

  Scalability is another important factor in the design of synchronization protocols. The computational complexity of synchronization algorithms becomes a critical problem as the number of sensors becomes very large. Besides, many other crucial MAC operations, such as multi-hop routing and network-configuration, highly depend on the network scalability as well.

C. Delay Components in Timing Message Delivery

The main role of time synchronization in a distributed network is to ensure a common timescale for all the network nodes, and to provide the right temporal coordination among all the nodes engaged in a collaborative and distributed interaction with the physical environment. Timing mismatch arises mainly from different setup times of nodes and time variations introduced by local oscillators running at different frequencies. Environmental variations, such as temperature and aging, also drive local clock oscillators run even unpredictably. All these uncertainties cause the local clocks of different nodes to drift away from each other over the course of a time interval.

Assume two nodes need to be synchronized. One of the node sends its current time to a neighboring node, if there is absolutely no delay in the message delivery, that neighboring node can immediately know the difference between its clock and its neighbor's clock. Unfortunately, in a real wireless network, various delays affect the message delivery, making time synchronization much more difficult than it seems to be. In general, a series of timing message transmissions is required to estimate the relative time offsets among nodes. In some sense, time synchronization in wireless sensor networks can be regarded as a process of removing the non-deterministic delays during timing message transmission over wireless channels.

There are a number of non-deterministic delays while transferring messages between nodes. *Kopetz* and *Ochsenreiter* for the first time analyzed the structure of message delays and characterized the delay components according to the process of message delivery [19]. The delay components in message delivery can be categorized into

1. Send Time: the time spent in building the message at the application layer including other delays introduced by the operating system when processing the

send request. The send time is nondeterministic and can be up to hundreds of milliseconds depending on the workload of the system.

2. Access Time: the waiting time for accessing the channel after reaching the MAC layer. This is the most significant factor and highly variable according to the specific MAC protocol. The access time is nondeterministic and varies from milliseconds up to seconds depending on the current network traffic.

3. Transmission Time: the time for transmitting a message at the physical layer. This delay can be estimated by the length of a message and the speed of radio in the medium and is in the order of tens of milliseconds.

4. Propagation Time: the actual time for a message to transmit from the sender to the receiver in a wireless channel. The propagation time is deterministic and less than one microsecond, which is almost negligible comparing with the other delay components.

5. Reception Time: the time required for receiving a message at the physical layer, which is the same as the transmission time. In some cases, this delay has been categorized as a part of the receive time to be presented next.

6. Receive Time: the time to construct and send a received message to the application layer at the receiver. It can be viewed as the corresponding component at the receiver side of the send time at the transmitter side, and can be time-varying due to the variable delays introduced by the operating system.

Note that the time delay in message transmission is also depending on other factors, such as hardware platform, error correction code, and modulation scheme. The estimated time delay discussed above in each component is based on the *MICA* platform [20], [21]. More detailed analysis can be found in [22]-[26].

CHAPTER III

FUNDAMENTAL APPROACHES TO TIME SYNCHRONIZATION

Time synchronization in wireless sensor networks can be achieved by transferring a group of timing messages to the target sensors. The timing messages contain the information about the time-stamps measured by the transmitting sensors. There exist two well-known approaches for time synchronization in wireless sensor networks, which are categorized as sender-receiver synchronization (SRS) and receiver-receiver synchronization (RRS). SRS is based on the traditional model of two-way message exchanges between a pair of nodes. For RRS, the nodes to be synchronized first receive a beacon packet from a common sender, then compare their receiving time readings of the beacon packet to compute the relative clock offset. Most of the existing time synchronization protocols rely on one of these two approaches. For instance, the Network Time Protocol (NTP) [5] and the Timing-sync Protocol for Sensor Networks (TPSN) [14] adopt SRS since they depend on a series of pairwise synchronizations that assume two-way timing message exchanges. Notice also that the Reference Broadcast Synchronization (RBS) protocol [27] relies on RRS since it requires pairs of message exchanges among children nodes (except the reference) to compensate their relative clock offsets.

Recently, a new approach for time synchronization, called the receiver-only synchronization (ROS), was proposed. It aims at minimizing the number of required timing messages and energy consumption during synchronization while preserving a high level of accuracy [28]. This approach can be used to achieve network-wide synchronization with much lesser timing messages than other well-known existing protocols such as TPSN and RBS.

Next we will present and analyze each of these synchronization approaches and

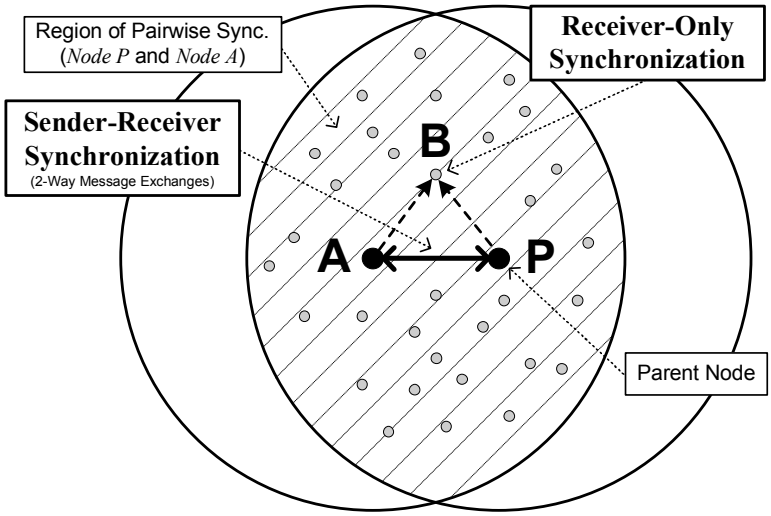

Fig. 1. Sender-receiver synchronization and receiver-only synchronization.

illustrate how the general design considerations can be resolved in these approaches. For all these approaches, we only present the underlying signaling mechanisms for performing pairwise synchronization, i.e., synchronizing a pair of nodes, since network-wide synchronization can be simply achieved by performing a group of pairwise synchronizations.

A. Sender-Receiver Synchronization

This approach is based on the classical two-way timing message exchange mechanism between two adjacent nodes. Consider a parent *Node P* and one of its children node *Node A* in Fig 1. The clock model for the two-way message exchange is depicted in Fig. 2, where $\theta_{\text{o}}^{(\text{AP})}$ denotes the clock offset between *Node A* and *Node P* and timing messages are assumed to be exchanged multiple (N) times [7], [14]. Here, the time stamps made during the ith message exchange $T_{1,i}^{(\text{A})}$ and $T_{4,i}^{(\text{A})}$ are measured by the local clock of *Node A*, and $T_{2,i}^{(\text{P})}$ and $T_{3,i}^{(\text{P})}$ are measured by the local clock of *Node P*, respectively. *Node A* transmits a synchronization packet, containing the value

of time stamp $T_{1,i}^{(\mathrm{A})}$ to *Node P*. *Node P* receives it at time $T_{2,i}^{(\mathrm{P})}$ and transmits an acknowledgement packet to *Node A* at $T_{3,i}^{(\mathrm{P})}$. This packet contains the value of time stamps $T_{1,i}^{(\mathrm{A})}$, $T_{2,i}^{(\mathrm{P})}$, and $T_{3,i}^{(\mathrm{P})}$. Then, *Node A* finally receives the packet at $T_{4,i}^{(\mathrm{A})}$.

As discussed before, packet delays can be characterized into several distinct components: send, access, transmission, propagation, and receive times. These delay components are divided into two parts: the fixed portion d and the variable portion X_i. The variable portion of delays depends on various network parameters (e.g., network status, traffic, etc.) and setup, and therefore no single delay model can be found to fit for every case. Thus far, several probability density function (PDF) models have been proposed for modeling random delays, the most widely deployed ones being Gaussian, Gamma, exponential and Weibull PDFs [27], [29], and [30]. The Gaussian delay model is appropriate if the delays are thought to be the addition of numerous independent random processes. In [27], the chi-squared test showed that the variable portion of delays can be modeled as Gaussian distributed random variables (RVs) with 99.8% confidence. On the other hand, a single-server M/M/1 queue can fittingly represent the cumulative link delay for point-to-point hypothetical reference connection, where the random delays are independently modeled as exponential RVs [31]. Thus, we assume the random portions of delays are either normal or exponentially distributed RVs.

Suppose that the clock frequencies of two nodes remain equal during the synchronization period, and both $X_i^{(\mathrm{AP})}$ and $X_i^{(\mathrm{PA})}$ are normal distributed RVs with mean μ and variance $\sigma^2/2$. From Fig. 2, $T_{2,i}^{(\mathrm{P})}$ and $T_{4,i}^{(\mathrm{A})}$ can be expressed as

$$T_{2,i}^{(\mathrm{P})} = T_{1,i}^{(\mathrm{A})} + \theta_{\mathrm{o}}^{(\mathrm{AP})} + d^{(\mathrm{AP})} + X_i^{(\mathrm{AP})}, \tag{3.1}$$

$$T_{4,i}^{(\mathrm{A})} = T_{3,i}^{(\mathrm{P})} + \theta_{\mathrm{o}}^{(\mathrm{PA})} + d^{(\mathrm{PA})} + X_i^{(\mathrm{PA})}, \tag{3.2}$$

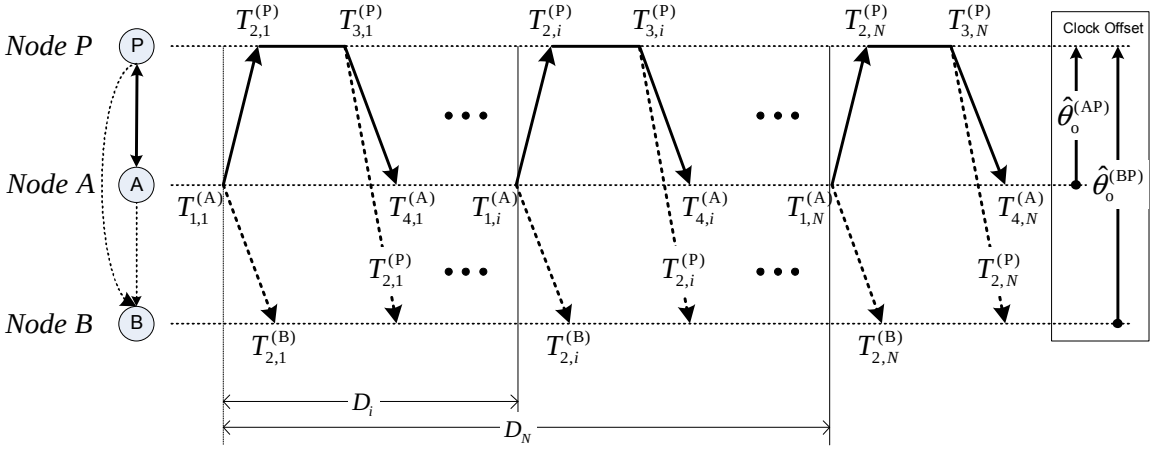

Fig. 2. Clock synchronization model of SRS and ROS.

where $\theta_{\rm o}^{(\rm PA)} = -\theta_{\rm o}^{(\rm AP)}$, and $d^{(\rm AP)}$ and $X_i^{(\rm AP)}$ denote the fixed and random portions of timing delays in the message transmissions from *Node A* to *Node P*, respectively. By defining the delays in up-link $U_i \triangleq T_{2,i}^{(\rm P)} - T_{1,i}^{(\rm A)}$ and down-link $V_i \triangleq T_{4,i}^{(\rm A)} - T_{3,i}^{(\rm P)}$, the ith delay observations corresponding to the ith timing message exchange are given by $U_i = \theta_{\rm o}^{(\rm AP)} + d^{(\rm AP)} + X_i^{(\rm AP)}$ and $V_i = \theta_{\rm o}^{(\rm PA)} + d^{(\rm PA)} + X_i^{(\rm PA)}$, respectively. Then, the likelihood function based on the observations $\{U_i\}_{i=1}^N$ and $\{V_i\}_{i=1}^N$ is given by

$$L(\theta_{\rm o}^{(\rm AP)}, \mu, \sigma^2) = (\pi\sigma^2)^{-\frac{N}{2}} e^{-\frac{1}{\sigma^2}\left[\sum_{i=1}^{N}(U_i - d^{(\rm AP)} - \theta_{\rm o}^{(\rm AP)} - \mu)^2 + \sum_{i=1}^{N}(V_i - d^{(\rm PA)} + \theta_{\rm o}^{(\rm AP)} - \mu)^2\right]},$$

where N stands for the number of message exchanges. Differentiating the log-likelihood function leads to

$$\frac{\partial \ln L(\theta_{\rm o}^{(\rm AP)})}{\partial \theta_{\rm o}^{(\rm AP)}} = -\frac{2}{\sigma^2}\sum_{i=1}^{N}[\theta_{\rm o}^{(\rm AP)} + d^{(\rm AP)} - d^{(\rm PA)} - (U_i - V_i)].$$

The fixed portions of delays are mainly determined by the propagation delays, and both up- and down-link channels have the same distance. Thus, the fixed portions of delays $d^{(\rm AP)}$ and $d^{(\rm PA)}$ are assumed to be equal, and is denoted by d for the rest of

this chapter. Indeed, the propagation delay is less than one microsecond for ranges under 300 meters, hence almost negligible when compared to other dominant delay components whose ranges are about hundreds of milliseconds [22]. The MLE of clock offset is given by

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{AP})} = \arg\max_{\theta_{\mathrm{o}}^{(\mathrm{AP})}} [\ln L(\theta_{\mathrm{o}}^{(\mathrm{AP})})] = \frac{\overline{U} - \overline{V}}{2}. \tag{3.3}$$

Thus, *Node A* can be synchronized to the parent node *Node P* by simply taking the difference of the average delay observations $\overline{U}$ and $\overline{V}$.

For exponential random delays $X_i^{(\mathrm{PA})}$ and $X_i^{(\mathrm{AP})}$ with the same mean λ, the likelihood function based on the observations $\{U_i\}_{i=1}^N$ and $\{V_i\}_{i=1}^N$ becomes

$$L(\theta_{\mathrm{o}}^{(\mathrm{AP})}, \lambda) = \lambda^{-2N} e^{-\frac{1}{\lambda}\sum_{i=1}^{N}[U_i+V_i-2d]} \cdot \prod_{i=1}^{N} I\left[U_i - \theta_{\mathrm{o}}^{(\mathrm{AP})} - d \geq 0,\ V_i + \theta_{\mathrm{o}}^{(\mathrm{AP})} - d \geq 0\right],$$

where $I(\cdot)$ stands for the indicator function (i.e., $I(\cdot)$ is 1 whenever its inner condition holds, otherwise being equal to 0). In [32], *Jeske* proved that the maximum likelihood estimator of $\theta_{\mathrm{o}}^{(\mathrm{AP})}$ exists when d is unknown and exhibits the same form as the estimator proposed in [33] and [34], namely

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{AP})} = \frac{\min\limits_{1\leq i\leq N} U_i - \min\limits_{1\leq i\leq N} V_i}{2}, \tag{3.4}$$

Notice from (3.3) and (3.4), it is clear that if only one round of message exchange is performed ($N = 1$), the MLE of clock offset for both exponential and Gaussian delay models become $\hat{\theta}_{\mathrm{o}}^{(\mathrm{AP})} = (U - V)/2$, which is exactly the same clock offset estimator adopted in [14].

Note that the clock offset between two nodes generally keeps increasing due to the difference of clock parameters of each oscillator. Thus, applying the clock skew correction mechanism increases the synchronization accuracy and guarantees the long-term reliability of synchronization. In Chapter V, we derive the joint clock offset and

skew estimators for the SRS approach. Besides, a family of robust and practical clock offset and skew estimators which do not require prior knowledge of d is also proposed in Chapter V.

B. Receiver-Only Synchronization

Due to the power constraint, the communication range of a sensor is strictly limited to a (radio-geometrical) circle whose radius depends on the transmission power (see Fig. 1). In this figure, every node within checked area (e.g., *Node B*) can receive messages from both *Node P* and *Node A*. Suppose that *Node P* is a parent (or reference) node, and *Node P* and *Node A* perform a pairwise synchronization using two-way timing message exchanges [14]. Then, all the nodes in the common coverage region of *Node P* and *Node A* (checked region) can receive a series of synchronization messages containing the information about the time stamps of the pairwise synchronization. Using this information, *Node B* can be also synchronized to the parent node *Node P* with no extra timing message transmissions. This approach is called receiver-only synchronization (ROS). In general, all the sensor nodes lying within the checked area can be synchronized by only receiving timing messages using ROS. Here, *Node P* and *Node A* can be regarded as super nodes since they provide synchronization beacons for all the nodes located in their vicinity.

In Fig. 1, consider an arbitrary node, say *Node B*, in the checked region. While *Node P* and *Node A* exchange time messages, *Node B* can overhear these time messages. Hence, *Node B* is capable of observing a set of time readings ($\{T_{2,i}^{(\mathrm{B})}\}_{i=1}^{N}$) at its local clock when it receives packets from *Node A* as depicted in Fig 2. Besides, *Node B* can also receive the information about a set of time stamps $\{T_{2,i}^{(\mathrm{P})}\}_{i=1}^{N}$ obtained by receiving the packets transmitted by *Node P*. Considering the effects of both clock

offset and skew, the reception time at *Node P* in the ith uplink message $T_{2,i}^{(\mathrm{P})}$ is given by

$$T_{2,i}^{(\mathrm{P})} = T_{1,i}^{(\mathrm{A})} + \theta_{\mathrm{o}}^{(\mathrm{AP})} + \theta_{\mathrm{s}}^{(\mathrm{AP})} \cdot (T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}) + d^{(\mathrm{AP})} + X_i^{(\mathrm{AP})}, \tag{3.5}$$

where $\theta_{\mathrm{s}}^{(\mathrm{AP})}$ stands for the relative clock skew between *Node A* and *Node P*. Likewise, the reception time at *Node B* in the ith uplink message $T_{2,i}^{(\mathrm{B})}$ can be represented by

$$T_{2,i}^{(\mathrm{B})} = T_{1,i}^{(\mathrm{A})} + \theta_{\mathrm{o}}^{(\mathrm{AB})} + \theta_{\mathrm{s}}^{(\mathrm{AB})} \cdot (T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}) + d^{(\mathrm{AB})} + X_i^{(\mathrm{AB})}, \tag{3.6}$$

where $\theta_{\mathrm{o}}^{(\mathrm{AB})}$ and $\theta_{\mathrm{s}}^{(\mathrm{AB})}$ stand for the relative clock offset and skew between *Node A* and *Node B*, $d^{(\mathrm{AB})}$ and $X_i^{(\mathrm{AB})}$ denote the fixed and random portions of timing delays in the message transmission from *Node A* to *Node B*, respectively. Here, $X_i^{(\mathrm{AB})}$ is assumed to be a normal distributed RV with mean μ and variance $\sigma^2/2$.

The linear regression technique can be applied to synchronize *Node B* and compensate the effects of the relative clock skew between *Node P* and *Node B*. Subtracting (3.6) from (3.5) gives

$$T_{2,i}^{(\mathrm{P})} - T_{2,i}^{(\mathrm{B})} = \theta_{\mathrm{o}}^{(\mathrm{BP})} + \theta_{\mathrm{s}}^{(\mathrm{BP})} \cdot (T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}) + d^{(\mathrm{AP})} - d^{(\mathrm{AB})} + X_i^{(\mathrm{AP})} - X_i^{(\mathrm{AB})}. \tag{3.7}$$

Since $d^{(\mathrm{AB})}$ and $d^{(\mathrm{AP})}$ are fixed values and $X_i^{(\mathrm{AB})}$ and $X_i^{(\mathrm{AP})}$ are normal distributed RVs, the noise component can be defined by $z[i] \triangleq \mu' + X_i^{(\mathrm{AP})} - X_i^{(\mathrm{AB})}$, where $\mu' \triangleq d^{(\mathrm{AP})} - d^{(\mathrm{AB})}$ and $z[i] \sim \mathcal{N}(\mu', \sigma^2)$. Let $x[i] \triangleq T_{2,i}^{(\mathrm{P})} - T_{2,i}^{(\mathrm{B})} - \mu'$ and $w[i] \triangleq z[i] - \mu'$, then the set of observed data can be written in matrix notation as follows:

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w},$$

where $\mathbf{x} = [x[1]\ x[2]\ \cdots\ x[N]]^T$, $\mathbf{w} = [w[1]\ w[2]\ \cdots\ w[N]]^T$, $\boldsymbol{\theta} = [\theta_{\mathrm{o}}^{(\mathrm{BP})}\ \theta_{\mathrm{s}}^{(\mathrm{BP})}]^T$, and

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & T_{1,2}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})} & \cdots & T_{1,N}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})} \end{bmatrix}^T.$$

Note that the noise vector $\mathbf{w} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ and the matrix $\mathbf{H}$ is the observation matrix whose dimension is $N \times 2$. From [35, Theorem 3.2, p. 44], the minimum variance unbiased (MVU) estimator for the relative clock offset and skew is given by $\hat{\boldsymbol{\theta}} = \mathbf{g}(\mathbf{x})$ where $\mathbf{g}(\mathbf{x})$ satisfies

$$\frac{\partial \ln p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{I}(\boldsymbol{\theta})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\theta}). \tag{3.8}$$

Since the noise vector $\mathbf{w}$ is zero mean and Gaussian distributed, from the results in [35, p. 85], the derivative of the log-likelihood function can be written as

$$\frac{\partial \ln p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\mathbf{H}^T\mathbf{H}}{\sigma^2}[(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x} - \boldsymbol{\theta}], \tag{3.9}$$

where $\mathbf{H}^T\mathbf{H}$ is assumed to be invertible. Therefore, comparing (3.8) with (3.9) yields

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}, \tag{3.10}$$

$$\mathbf{I}(\boldsymbol{\theta}) = \frac{\mathbf{H}^T\mathbf{H}}{\sigma^2}, \tag{3.11}$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the Fisher information matrix. After some mathematical manipulations, the joint clock offset and skew estimator can be expressed as

$$\begin{bmatrix} \hat{\theta}_{\mathrm{o}}^{(\mathrm{BP})} \\ \hat{\theta}_{\mathrm{s}}^{(\mathrm{BP})} \end{bmatrix} = \frac{1}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \begin{bmatrix} \sum_{i=1}^{N} D_i^2 \sum_{i=1}^{N} x[i] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} [D_i \cdot x[i]] \\ N\sum_{i=1}^{N} [D_i \cdot x[i]] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} x[i] \end{bmatrix}, \tag{3.12}$$

where $D_i \triangleq T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}$. The Cramer-Rao lower bound (CRB) can be obtained by inverting the Fisher information matrix $\mathbf{I}(\boldsymbol{\theta})$. From (3.11), the Fisher information matrix is given by

$$\mathbf{I}(\boldsymbol{\theta}) = \frac{1}{\sigma^2} \begin{bmatrix} N & \sum_{i=1}^{N} D_i \\ \sum_{i=1}^{N} D_i & \sum_{i=1}^{N} D_i^2 \end{bmatrix}.$$

Then, inverting $\mathbf{I}(\boldsymbol{\theta})$ yields

$$\mathbf{I}^{-1}(\boldsymbol{\theta}) = \frac{\sigma^2}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \begin{bmatrix} \sum_{i=1}^{N} D_i^2 & -\sum_{i=1}^{N} D_i \\ -\sum_{i=1}^{N} D_i & N \end{bmatrix}. \tag{3.13}$$

Hence, from (3.13), the CRBs for the relative clock offset and skew become

$$\mathrm{var}(\hat{\theta}_{\mathrm{o}}^{(\mathrm{BP})}) \geq \frac{\sigma^2 \sum_{i=1}^{N} D_i^2}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \tag{3.14}$$

and

$$\mathrm{var}(\hat{\theta}_{\mathrm{s}}^{(\mathrm{BP})}) \geq \frac{\sigma^2 N}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2}. \tag{3.15}$$

Notice further that the regularity conditions for the CRBs hold:

$$\begin{aligned} E\left[\frac{\partial \ln p(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] &= \begin{bmatrix} E\left[\frac{\partial \ln p(\mathbf{x};\boldsymbol{\theta})}{\partial \theta_{\mathrm{o}}^{(\mathrm{BP})}}\right] \\ E\left[\frac{\partial \ln p(\mathbf{x};\boldsymbol{\theta})}{\partial \theta_{\mathrm{s}}^{(\mathrm{BP})}}\right] \end{bmatrix} \\ &= \begin{bmatrix} E\left[\frac{1}{\sigma^2}\sum_{i=1}^{N}\left[x[n] - \theta_{\mathrm{o}}^{(\mathrm{BP})} - \theta_{\mathrm{s}}^{(\mathrm{BP})}\cdot D_i\right]\right] \\ E\left[\frac{1}{\sigma^2}\sum_{i=1}^{N}\left\{\left[x[n] - \theta_{\mathrm{o}}^{(\mathrm{BP})} - \theta_{\mathrm{s}}^{(\mathrm{BP})}\cdot D_i\right]\cdot D_i\right\}\right] \end{bmatrix} = \mathbf{0}. \end{aligned}$$

Consequently, using the results in (3.12), *Node B* can be synchronized to *Node P*. Likewise, all the other nodes in the checked region in Fig. 1 can be simultaneously synchronized to the parent node *Node P* without any additional timing message transmissions, thus saving a significant amount of energy.

C. Receiver-Receiver Synchronization

Receiver-receiver synchronization is an approach to synchronize a set of children nodes who receive the beacon messages from a common sender (a reference or parent node).

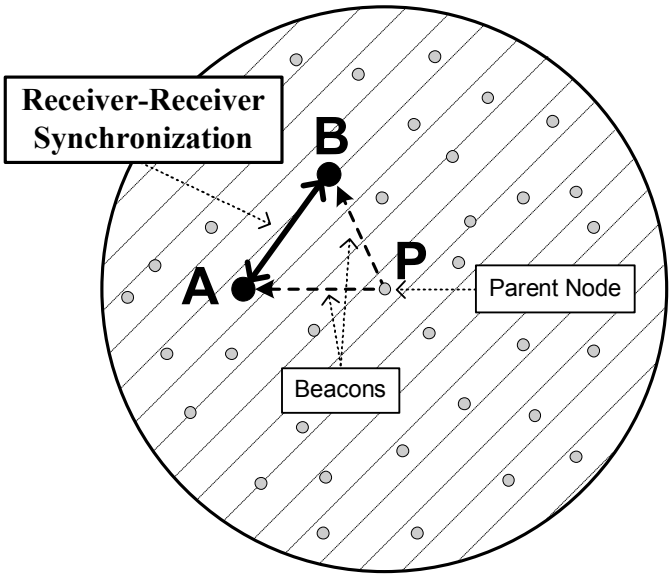

Fig. 3. Receiver-receiver synchronization.

Consider a parent (reference) node P and arbitrary nodes A and B, which locate within the communication range of the parent node in Fig. 3. Suppose, in Fig. 4 both *Node A* and *Node B* receive the i^{th} beacon from *Node P* at time instants $T_{2,i}^{(\mathrm{A})}$ and $T_{2,i}^{(\mathrm{B})}$ of their local clocks, respectively. Nodes A and B record the arrival time of the broadcast packet according to their own timescales and then exchange their time-stamps. Suppose $X_i^{(\mathrm{PA})}$ denotes the nondeterministic delay components (random portion of delays) and $d^{(\mathrm{PA})}$ denotes the deterministic delay component (propagation delay) from *Node P* to *Node A*, then $T_{2,i}^{(\mathrm{A})}$ can be written as

$$T_{2,i}^{(\mathrm{A})} = T_{1,i} + d^{(\mathrm{PA})} + X_i^{(\mathrm{PA})} + \theta_{\mathrm{o}}^{(\mathrm{PA})} + \theta_{\mathrm{s}}^{(\mathrm{PA})} \cdot (T_{1,i} - T_{1,1}), \tag{3.16}$$

where $T_{1,i}$ is the transmission time at the reference node, $\theta_{\mathrm{o}}^{(\mathrm{PA})}$ and $\theta_{\mathrm{s}}^{(\mathrm{PA})}$ are the clock offset and skew of *Node A* with respect to the reference node, respectively. Similarly, we can decompose the arrival time at *Node B* as

$$T_{2,i}^{(\mathrm{B})} = T_{1,i} + d^{(\mathrm{PB})} + X_i^{(\mathrm{PB})} + \theta_{\mathrm{o}}^{(\mathrm{PB})} + \theta_{\mathrm{s}}^{(\mathrm{PB})} \cdot (T_{1,i} - T_{1,1}), \tag{3.17}$$

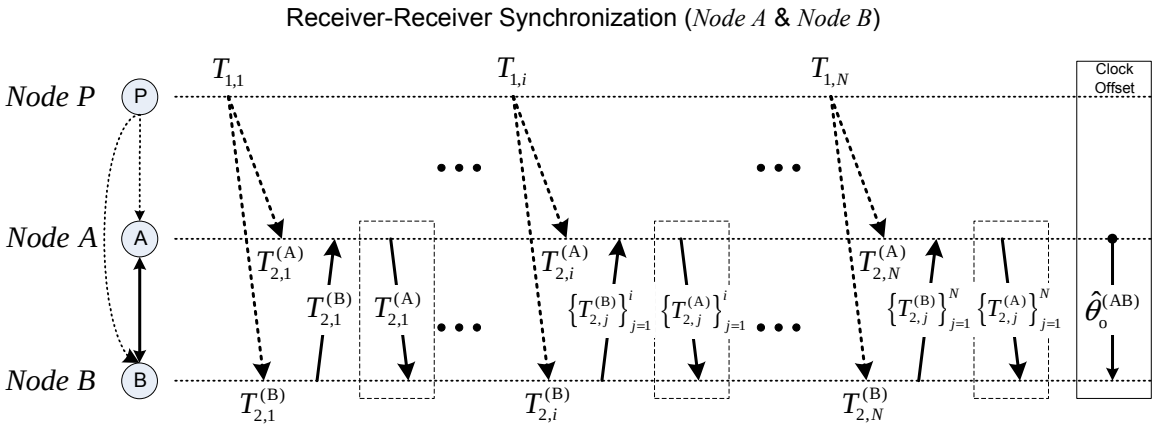

Fig. 4. Clock synchronization model of RRS.

where $d^{(\mathrm{PB})}$, $X_i^{(\mathrm{PB})}$, $\theta_{\mathrm{o}}^{(\mathrm{PB})}$, and $\theta_{\mathrm{s}}^{(\mathrm{PB})}$ stand for the propagation (fixed) delay, random portion of delays, clock offset and skew of *Node B* with respect to the reference node, respectively.

Subtracting (3.17) from (3.16), we obtain

$$T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} = \theta_{\mathrm{o}}^{(\mathrm{BA})} + \theta_{\mathrm{s}}^{(\mathrm{BA})} \cdot (T_{1,i} - T_{1,1}) + d^{(\mathrm{PA})} - d^{(\mathrm{PB})} + X_i^{(\mathrm{PA})} - X_i^{(\mathrm{PB})} \qquad (3.18)$$

where $\theta_{\mathrm{o}}^{(\mathrm{BA})} \triangleq \theta_{\mathrm{o}}^{(\mathrm{PA})} - \theta_{\mathrm{o}}^{(\mathrm{PB})}$ and $\theta_{\mathrm{s}}^{(\mathrm{BA})} \triangleq \theta_{\mathrm{s}}^{(\mathrm{PA})} - \theta_{\mathrm{s}}^{(\mathrm{PB})}$ are the relative clock offset and skew between *Node A* and *Node B* at the time they receive the i^{th} broadcast packet from the reference node, respectively. Here, we assume these random portions of delays $X_i^{(\mathrm{PA})}$ and $X_i^{(\mathrm{PB}}$ are normal distributed RVs with mean μ and variance $\sigma^2/2$. Indeed, (3.18) assumes exactly the same form as (3.7). Hence, the same steps can be applied to derive the joint clock offset and skew estimator for ROS. More specifically, let the noise component $z[i] \triangleq \mu' + X_i^{(\mathrm{BA})}$, where $\mu' \triangleq d^{(\mathrm{PA})} - d^{(\mathrm{PB})}$ and $z[i] \sim \mathcal{N}(\mu', \sigma^2)$. Let also define $x[i] \triangleq T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} - \mu'$ and $w[i] \triangleq z[i] - \mu'$. Using similar steps as in ROS, it is straightforward to show that the same form of the joint clock offset and skew estimator (3.12) can be also applied to RRS. Consequently, there is no difference between ROS and RRS with regard to the accuracy of synchronization

since the effects of random delays are the same. Likewise, the CRB for RRS can also be obtained using a similar procedure as in ROS. When there is no relative clock skew ($\theta_{\mathrm{s}}^{(\mathrm{BA})} = 0$), it is straightforward to show that the maximum likelihood estimator of the relative clock offset $\hat{\theta}_{\mathrm{o}}^{(\mathrm{BA})}$ becomes

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{BA})} = \frac{1}{N} \sum_{i=1}^{N} \left[T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} \right], \tag{3.19}$$

which is the equivalent to the estimator presented in [27].

The main benefit of this approach is that all non-deterministic delay components on the transmitter side (send time and access time) are eliminated. Thus, a high degree of synchronization accuracy can be achieved using this approach.

D. Comparisons

Sender-receiver synchronization can be directly time-stamped at the physical layer so as to eliminate the effects of delay components related to the operating system. Hence, it significantly mitigates the uncertainty of timing delays in message delivery. In contrast, receiver-receiver synchronization removes the effect of nondeterministic delay components, such as send and access times, on the receiver side. Experimental results using the Berkeley mote platform in [14] claim that sender-receiver synchronization outperforms receiver-receiver synchronization in terms of synchronization accuracy (errors) by roughly two times. However, it is arguable since performance depends on a variety of different factors, such as the network platform and setup, channel status, and estimation schemes.

Receiver-only synchronization aims at minimizing the overall energy consumption in synchronization. In this approach, a number of sensor nodes can be synchronized without any message transmission, i.e., they can be synchronized by only receiving

timing messages between pairs of nodes. Although there will be no gain regarding the synchronization accuracy compared with the other approaches, receiver-only synchronization significantly reduces the overall network-wide energy consumption by decreasing the number of required timing messages in synchronization.

CHAPTER IV

TIME SYNCHRONIZATION PROTOCOLS

Thus far, a number of protocols have been suggested to solve the problem of time synchronization in distributed networks. For general computer networks, NTP has been adopted as the standard time synchronization scheme of the Internet [5]. Although NTP was shown to perform well in computer networks, it is not directly applicable to wireless sensor networks due to the unique challenges sensor networks face: limited power resources, wireless channel conditions, dynamic topology changes, etc., (recall also the design considerations presented in Chapter II). NTP enjoys unlimited (or rechargeable) energy resources and a relatively static topology in computer networks. However, these are not available in sensor networks. Therefore, different types of time synchronization protocols have been proposed to meet the design requirements of wireless sensor networks [3].

Ideally, a time synchronization protocol should be able to work optimally in terms of all the design requirements imposed on time synchronization, which are energy efficiency, scalability, precision, security, reliability, and robustness to network dynamics. However, the complex nature of wireless sensor networks makes it very difficult to optimize the protocol with respect to all these requirements simultaneously. Due to the tradeoffs in satisfying these requirements, each protocol is designed to put distinct emphases on different requirements.

Assuming various criteria, time synchronization protocols can be categorized into different classes:

- **Master-Slave vs. Peer-to-Peer**
  - *Master-Slave*: where first a tree-like network hierarchy is arranged, and

upon the completion of this arrangement only the connected nodes in the hierarchy synchronize with each other.

  - *Peer-to-Peer*: where any pair of nodes in the network can synchronize with each other.

- **Clock Correcting vs. Untethered Clock**

  - *Clock Correcting*: where the clock function in memory is modified after each run of the time synchronization process.

  - *Untethered Clock*: where every node maintains its own clock as it is, and keeps a time-translation table relating its clock to other nodes' clocks; thus, instead of updating its clock constantly, each node translates the time information in the data packets coming from other nodes to its own clock by using the time-translation table.

- **Synchronization Approach**

  - *Sender-Receiver*: where one of two nodes, which are synchronizing with each other, sends a time-stamp message while the other one receives it.

  - *Receiver-Receiver*: where a reference node transmits synchronization-signals and two synchronizing nodes receive these signals and record the time of receptions (time-stamps).

  - *Receiver-Only*: where a group of nodes can be simultaneously synchronized by only listening to the message exchanges of a pair of nodes.

- **Pairwise Synchronization vs. Network-Wide Synchronization**

  - *Pairwise synchronization*: where the protocols are primarily designed to synchronize two nodes, although they usually can be extended to handle

synchronization of a group of nodes.

– *Network-wide synchronization*: where the protocols are primarily designed to synchronize a large number of nodes in the network.

Additional classifications could be found in [7]. In the following, we will summarize the existing time synchronization protocols based on the last category.

A. Pairwise Synchronization

1. Timing-Sync Protocol for Sensor Networks (TPSN)

TPSN [14] uses the two-way message exchange mechanism, as discussed in the sender-receiver synchronization approach (described in Chapter III), to achieve the synchronization between two nodes. With only one round of message exchanges, and without any statistical model on the variable delay components $X_i^{(\mathrm{AP})}$ and $X_i^{(\mathrm{PA})}$ in (3.1) and (3.2), a simple estimate for $\theta_{\mathrm{o}}^{(\mathrm{AP})}$ is proposed in [14] as

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{AP})} = \frac{U_i - V_i}{2}, \tag{4.1}$$

where $U_i \triangleq T_{2,i}^{(\mathrm{P})} - T_{1,i}^{(\mathrm{A})}$ and $V_i \triangleq T_{4,i}^{(\mathrm{A})} - T_{3,i}^{(\mathrm{P})}$. Notice that in the original form of TPSN, it does not estimate clock skew, therefore, frequent application of TPSN is needed to keep the clock offset between two nodes under a certain limit.

Assume the clock offset $\theta_{\mathrm{o}}^{(\mathrm{AP})}$ is constant for N rounds of message exchanges. If $X_i^{(\mathrm{AP})}$ and $X_i^{(\mathrm{PA})}$ are exponentially distributed with the same unknown mean λ and when $d \triangleq d^{(\mathrm{AP})} = d^{(\mathrm{PA})}$ is unknown, it is proved in [32] that the ML estimator of $\theta_{\mathrm{o}}^{(\mathrm{AP})}$ is given by

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{AP})} = \frac{\min_{1\le i\le N} U_i - \min_{1\le i\le N} V_i}{2}. \tag{4.2}$$

On the other hand, with $X_i^{(\mathrm{AP})}$ and $X_i^{(\mathrm{PA})}$ being modeled as independent and normally

distributed RVs with the same mean μ and variance $\sigma^2/2$, the Maximum Likelihood (ML) estimate for $\theta_{\rm o}^{(\rm AP)}$ takes the equation (derived in Chapter III)

$$\hat{\theta}_{\rm o}^{(\rm AP)} = \frac{\frac{1}{N}\sum_{i=1}^{N} U_i - \frac{1}{N}\sum_{i=1}^{N} V_i}{2}. \tag{4.3}$$

From (4.1), (4.2) and (4.3), it is clear that if only one round of message exchange is performed, the TPSN presented in (4.1) is the ML estimator under both exponential and Gaussian delay models.

In Chapter V, we propose a practical joint clock offset and skew correction scheme to guarantee the long-term stability of synchronization for TPSN. Moreover, the joint offset and skew ML estimators for TPSN under Gaussian delay assumption are also derived in Chapter V.

2. Tiny-Sync and Mini-Sync

Tiny-sync and Mini-sync [36] are two lightweight clock synchronization protocols that also use the two-way message exchanges. Suppose that *Node A* and *Node P* exchange timing messages like in Fig. 5. This figure shows the effect of clock offset ($\theta_{\rm o}$) and skew ($\theta_{\rm s}$) on timing message exchanges between two nodes. Without loss of generality, the reference time $T_{1,1}^{(\rm A)}$ is set to be zero. Here, the time stamp at *Node P* in the ith uplink message $T_{2,i}^{(\rm B)}$ is given by

$$\begin{aligned} T_{2,i}^{(\rm P)} &= T_{1,i}^{(\rm A)} + \theta_{\rm o}^{(\rm AP)} + \theta_{\rm s}^{(\rm AP)}(T_{1,i}^{(\rm A)} + d + X_i^{(\rm AP)}) + d + X_i^{(\rm AP)} \\ &= (1+\theta_{\rm s}^{(\rm AP)})(T_{1,i}^{(\rm A)} + d + X_i^{(\rm AP)}) + \theta_{\rm o}^{(\rm AP)}, \end{aligned} \tag{4.4}$$

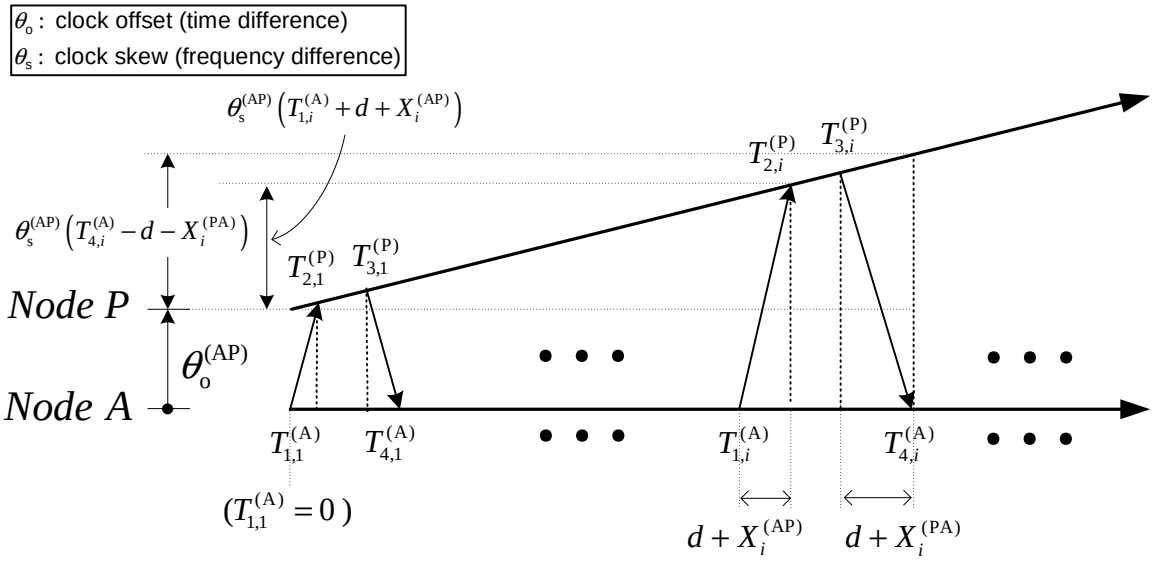

Fig. 5. Linear clock skew model for message exchanges.

where the term $\theta_{\rm s}^{\rm (AP)}(T_{1,i}^{\rm (A)} + d + X_i^{\rm (AP)})$ is due to the effect of clock skew. Similarly, the time stamp at *Node P* in the ith downlink message $T_{3,i}^{\rm (P)}$ takes the equations

$$\begin{aligned} T_{3,i}^{\rm (P)} &= T_{4,i}^{\rm (A)} + \theta_{\rm o}^{\rm (AP)} + \theta_{\rm s}^{\rm (AP)}(T_{4,i}^{\rm (A)} - d - X_i^{\rm (PA)}) - d - X_i^{\rm (PA)} \\ &= (1 + \theta_{\rm s}^{\rm (AP)})(T_{4,i}^{\rm (A)} - d - X_i^{\rm (PA)}) + \theta_{\rm o}^{\rm (AP)}, \end{aligned} \tag{4.5}$$

where the term $\theta_{\rm s}^{\rm (AP)}(T_{4,i}^{\rm (A)} - d - X_i^{\rm (PA)})$ is again due to the effect of clock skew.

This protocol assumes that *Node P* reply to *Node A* immediately after receiving the message, i.e., $T_{2,i}^{\rm (P)} = T_{3,i}^{\rm (P)}$. Suppose the clocks between *Node A* and *Node P* are linearly related, from (5.9) and (5.10), we have

$$\begin{aligned} \frac{T_{2,i}^{\rm (P)} - \theta_{\rm o}^{\rm (AP)}}{1 + \theta_{\rm s}^{\rm (AP)}} &= T_{1,i}^{\rm (A)} + d + X_i^{\rm (AP)}, \\ \frac{T_{2,i}^{\rm (P)} - \theta_{\rm o}^{\rm (AP)}}{1 + \theta_{\rm s}^{\rm (AP)}} &= T_{4,i}^{\rm (A)} - d - X_i^{\rm (PA)}. \end{aligned}$$

Since d, $X_i^{\rm (AP)}$ and $X_i^{\rm (PA)}$ are all non-negative, defining $\theta'_{\rm s} \triangleq 1/(1 + \theta_{\rm s}^{\rm (AP)})$ and $\theta'_{\rm o} \triangleq$

$\theta_{\mathrm{o}}^{(\mathrm{AP})}/(1+\theta_{\mathrm{s}}^{(\mathrm{AP})})$, we obtain

$$T_{1,i}^{(\mathrm{A})} \leq \theta'_{\mathrm{s}} T_{2,i}^{(\mathrm{P})} + \theta'_{\mathrm{o}} \leq T_{4,i}^{(\mathrm{A})}. \tag{4.6}$$

The 3-tuple of timestamps ($T_{1,i}^{(\mathrm{A})}, T_{2,i}^{(\mathrm{P})}$, and $T_{3,i}^{(\mathrm{A})}$) is called a data point. With N message exchanges, the goal is to find θ'_o and θ'_{s} such that they satisfy (4.6) for $1 \leq i \leq N$. In general, this is a linear programming problem and there are an infinite number of solutions for this problem [37]. Although more timestamps would generate tighter bounds on θ'_{o} and θ'_{s}, unfortunately, at the same time, the computational and storage requirements of the linear programming approach also increases. Thus, such an approach does not appear suitable to be implemented in wireless sensor nodes, which have strictly limited memory and computing resources.

Tiny-sync and Mini-sync tackle the problem as finding the best-fit line that lies between the bound sets defined by the data points. Based on the observation that not all data points are useful, Tiny-sync preserves only four constraints (the ones that yield the best bounds on the estimate) out of all data points. This results in a very efficient algorithm. However, it is shown by a counterexample [36] that this scheme does not always produce the optimal solution since some data points are considered useless and discarded at a certain time, a step which would actually might provide a better bound if it is properly considered with another data point that is yet to come.

Mini-sync is an improved version of Tiny-sync in the sense that it finds the optimal solution with increased complexity (but still with lesser complexity than the linear programming approach). Mini-sync basically uses an additional criterion to determine whether the data point can be safely discarded.

3. Reference Broadcast Synchronization (RBS)

RBS [27] is based on the RRS approach discussed in Chapter III. Let the time-stamps recorded at *Node A* and *Node B* for receiving the i^{th} common packet be denoted as $T_{2,i}^{(\mathrm{A})}$ and $T_{2,i}^{(\mathrm{B})}$, respectively. The estimate of the clock offset between *Node A* and *Node B* is proposed in [27] as

$$\hat{\theta}_{\mathrm{o}}^{(\mathrm{BA})} = \frac{1}{N}\sum_{i=1}^{N}\left[T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})}\right], \tag{4.7}$$

where N stands for the total number of common packets received by *Node A* and *Node B*. We have shown in Chapter III that the above estimator is actually the ML estimator for the clock offset, assuming the random portions of the delays in message deliveries are Gaussian distributed RVs, and there is no clock skew. When there is a clock skew between *Node A* and *Node B*, least-squares linear regression is proposed in [27] to estimate the clock skew.

The main advantage of RBS is that by comparing the time stamps of a common packet at two different nodes, it removes the largest sources of non-deterministic error (send time and access time) from the transmission path. Thus, RBS provides a high degree of synchronization accuracy. Note also that RBS can be applied to commodity hardware and existing software in sensor networks as it does not need access to the low levels of the operating system.

Under the setting that a sensor node observes and synchronizes to a broadcast clock, [38] derives the ML estimator for clock offset and skew with the broadcast message delay being modeled as uniformly distributed RVs. It is shown that the ML estimate in this case is generally not unique. Furthermore, the support of likelihood function is not convex which leaves out the possibility of taking the mean of all equally likely solutions. This motivated [38] to consider the linear estimator for the clock

offset and skew. Under the same setting, [39] derives the joint ML clock offset and skew estimator with the assumption that the broadcast message delays are modeled as exponentially distributed RVs. It is shown in [39] that a unique joint ML clock offset and skew estimate exists under certain conditions, as opposed to the case of uniformly distributed delay. Furthermore, Gibbs sampler was introduced in [39] to further enhance the performance of the joint ML estimator.

4. Flooding Time Synchronization Protocol (FTSP)

In [22], it is argued that if one can time-stamp the message at MAC layer, this immediately eliminates three sources of delay uncertainties: transmit, access and receive times. In this case, the main delivery delay comes from transmission and reception times at the radio chips (see Chapter II). These delays can be further decomposed into 1) interrupt handling time, which is the delay between the radio chip raising and the microcontroller responding to an interrupt; 2) encoding time, which is the time it takes for the radio chip to encode and transform the message into a radio wave; 3) decoding time, which is the time for the radio chip at the receiver to transform the radio wave back into binary data; and 4) byte alignment time, which is the delay at the receiver to synchronize with the byte boundary at the physical layer.

FTSP [22] uses a single broadcasted message to establish synchronization points between sender and receivers, while eliminating the jitter of interrupt handling and encoding/decoding times by utilizing multiple MAC layer time stamps both on the sender and receiver sides. Furthermore, the skew of the clock between sender and receiver is estimated using multiple messages and linear regression.

B. Network-Wide Synchronization

Until this point, we have only described the time synchronization between two neighboring sensor nodes. In this section, we will discuss protocols for network-wide synchronization.

1. Extension of TPSN

In order to establish a global timescale for all the nodes in the sensor field based on TPSN, [14] proposes to create a hierarchical structure (spanning tree) in the network (named level discovery phase) before pairwise synchronization being performed between adjacent levels (named synchronization phase). The level discovery phase consists of the following steps: 1) select a root node using an appropriate leader election algorithm and assign a 0-level to the root node; 2) the root node broadcast a level discovery packet (LDP) containing the identity and the level of packet; 3) every node who receives a LDP assigns its level to a level greater (by one) than that of the received packet and sends a new level discovery packet attaching its own level (once being assigned a level, a node neglects future packets requesting level discovery to avoid flooding congestion); 4) repeat step 3) until every node in the network successfully assigns a level.

After the spanning tree is formed, the root node initiates the synchronization phase by synchronizing all the nodes in level 1. Next, the nodes in level 1 synchronize with the nodes in level 2, and so on, until all the nodes have been synchronized. Notice that the synchronization error of a node with respect to the root node is a non-decreasing function of the hop distance because the random signal errors over each hop add up. A number of different searching algorithms can be considered in the construction of the spanning tree. For instance, Van Greunen and Rabaey suggested

some preliminary ideas on constructing spanning trees with low depth in order to improve the accuracy of synchronization [18].

2. Lightweight Time Synchronization (LTS)

Also based on two-way message exchanges, [18] proposes two network-wide synchronization protocols. The first one is called centralized multi-hop LTS, which is basically the same protocol as the extension of TPSN discussed above. The other one is called distributed multi-hop LTS. This distributed LTS algorithm moves the re-synchronization from the root node to the nodes that needs re-synchronization. When a *Node A* determines that it needs to be re-synchronized, it will send a re-synchronization request to the root node. In order for *Node A* to re-synchronize, all nodes along the routing path from the root node to *Node A* will be synchronized in a pairwise fashion. In case that clock skews are bounded, LTS provides an alternative approach with low-complexity and high-efficiency. In [40] and [41], the probabilistic approaches have been developed and extended. Besides, performance bounds under various different assumptions have been analyzed in [42]-[44].

3. Extension of RBS

The RBS protocol discussed in the above subsection can only synchronize a set of nodes that lie within a single broadcast domain. In order to synchronize a large sensor network, [27] proposes to use *gateway* nodes for converting timestamps from one neighborhood's timebase to another. The idea is illustrated in Fig. 6. *Nodes P1* and *P2* send out synchronization beacons, and they create two overlapping neighborhoods, where *Node B* lie in the overlapping area. Since *Node A* and *Node B* lie within the same neighborhood, their clock relationship (i.e., clock offset and skew) can be established from the *Node P1*'s reference broadcast. Similarly the clock relationship

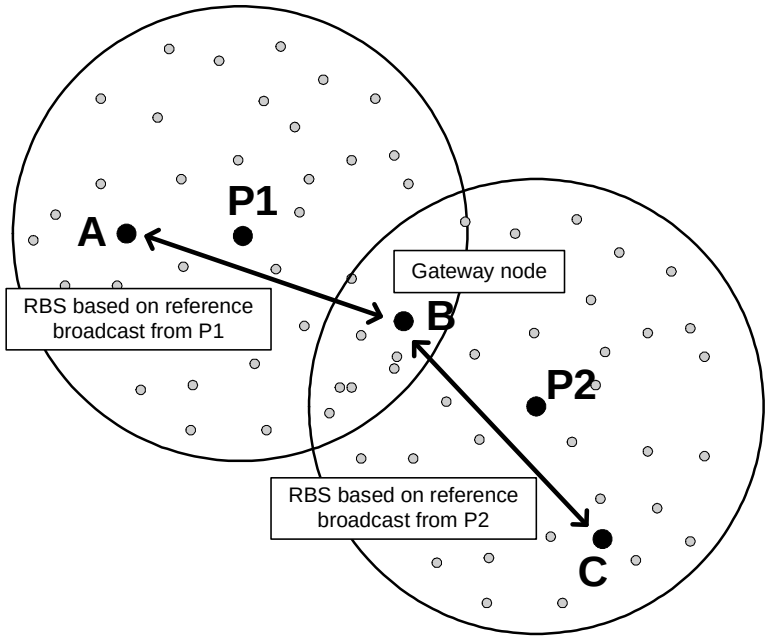

Fig. 6. Extension of RBS to multi-hop.

between *Node B* and *Node C* can be established from the *Node P2*'s reference broadcast. Therefore, the clock relationship between *Node A* and *Node C* can be computed with *Node B* acting as a gateway.

4. Extension of FTSP

FTSP can be extended to network-wide synchronization in a straightforward manner. First, a root node, to which the whole network is being synchronized, is elected by the network. Nodes that are within the broadcast radius of the root node can receive time-stamped messages from the root node. They then estimate the offset and skew of their own local clocks, thus synchronizing with the root node. The newly synchronized nodes can then broadcast synchronization messages to other nodes in the network. The advantage of this flooding process is that it begins with the root node, and there is no need to have a level hierarchy, as opposed to TPSN.

5. Pairwise Broadcast Synchronization

In this dissertation, we propose the Pairwise Broadcast Synchronization (PBS) protocol, which employs both sender-receiver and receiver-only synchronization approaches to achieve network-wide synchronization with high energy efficiency [28]. In PBS, a number of sensor nodes can be synchronized by only overhearing timing messages being exchanged between pairs of nodes, which significantly reduces the overall energy consumption by decreasing the number of required timing messages in synchronization. PBS requires a much smaller number of timing messages than other existing protocols such as RBS, TPSN, and FTSP, and its benefits remarkably increase as the sensors are more densely deployed.

In fact, a similar concept of combining the merits of both RRS and SRS approaches has been applied in TDP. TDP elects the diffusion leaders in every level of the network and the selected leaders successively broadcast synchronization messages. However, unlike TDP, the proposed PBS selects the best set of synchronization pairs to minimize the number of overall timing messages and energy consumption, while TDP is based on drastically different election criteria: the balance of work loads and the clock stability. Besides, in TDP, there were no concerns about the optimum number of diffusion leaders in terms of energy-efficiency and how to guarantee the network-wide synchronization. Chapter VI illustrates and analyzes the proposed PBS in detail.

6. Time Diffusion Protocol (TDP)

TDP [45] is a protocol enabling the sensor network to reach an equilibrium time with the clocks of individual sensors within a small time deviation from the equilibrium time. The protocol can be understood as periodically applying three phases: 1)

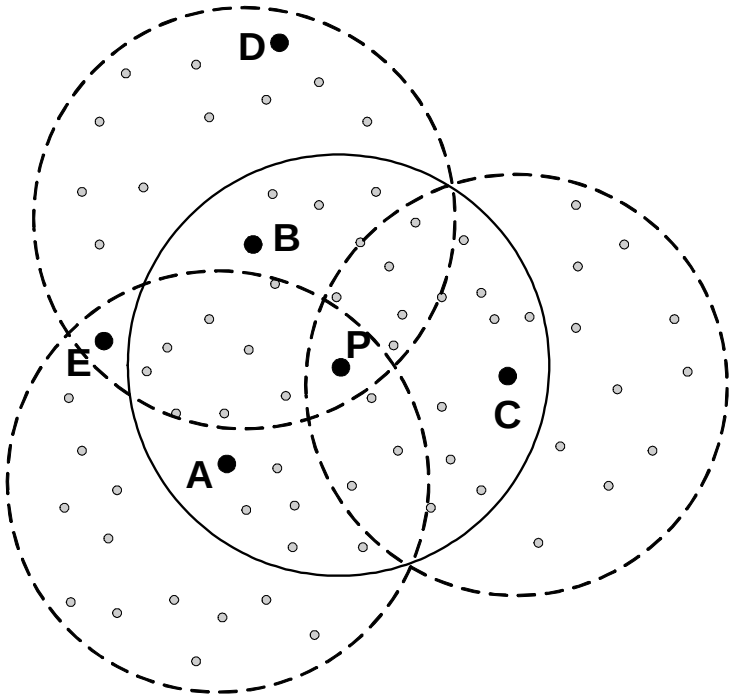

Fig. 7. Time-diffusion synchronization protocol.

Election of master/diffused leader nodes, 2) Time diffusion procedure and 3) Peer evaluation procedure. It is shown analytically in [45] that the TDP enables the clocks in the whole network to converge to a unique value.

In the first phase, master nodes are elected in the sensor field. The election criteria include the quality of clock and the energy resources of a particular node. Referring to Fig. 7, assume that *Node P* is elected to be the master node (here we illustrate the concept with one master node, while in more complicated networks, more than one master node might be possible). *Node P* then sends a number of time-stamped messages to its neighbors. Once the neighbors receive the messages, they self-determine if they would become diffused leader nodes, based on the results of the last round Peer evaluation procedure (the third phase). In Fig. 7, *Nodes A*, *B* and *C* are the elected diffused leader nodes. The elected diffused leader nodes respond

to the master node, thus enabling the master node to measure the average and the standard deviation of the round-trip delay from its neighbors. At the same time, the diffused leader nodes start sending messages to their own neighbors to measure the mean and standard deviation of round-trip delay to their neighbors. The process is repeated until all the nodes have been covered.

In the second phase, the time information from the master node will be diffused (with the help of diffused leader nodes) to all the nodes in the network. The diffusion procedure takes place according to the following sequence of events. First, the master node sends a time-stamped message containing the standard deviation of the round-trip delay to its neighbors. Before transmission, the time-stamp of the message is adjusted with half of the measured average round-trip delay (from the first phase) to account for the message delivery delay to its neighbors. Once the diffused leader nodes receive the time-stamped message, they set their clock according to the received time-stamp and then broadcast their own time-stamped messages, containing their measured standard derivations of the round-trip times to their neighbors. Again, before transmission of the messages, the time-stamps have to be adjusted with half of the measured average round-trip delay to their neighbors. For nodes that are not diffused leaders, if they only receive a message from one diffused leader node (e.g., *Node D* in Fig. 7), they just set their clock according to the time-stamp they received. For the nodes that have received more than one time-stamped messages originated from different diffused leader nodes (e.g., *Node E* in Fig. 7), they will use the standard deviations as weightings (the smaller the deviation, the larger the weighting) to combine the clock values and set their clocks according to the result.

The purpose of the third phase is to allow the sensor nodes to evaluate the stability of their local clock. First, the elected master nodes broadcast a number of time-stamped messages. The neighbor nodes receiving these messages calculate

the 2-sample *Allan* variance [45] of the local clock from the clock of the master nodes and send back these calculated *Allan* variances to the master nodes. Then the master nodes compute the average of all the *Allan* variances they received and send the result back to their neighbor nodes [46]. By this procedure, all the neighbor nodes can evaluate the quality of their clocks with respect to those of their neighbors by comparing their calculated *Allan* variance with the average value. The above procedure is repeated, but with the elected diffused leader nodes broadcasting the time-stamped messages.

7. Synchronous and Asynchronous Diffusion Algorithms

In [47], two diffusion algorithms are proposed. The first one is called rate-based synchronous diffusion algorithm. The idea behind this algorithm is that in order for a network to achieve an equilibrium time, the clock at *Node* i, denoted as c_i, should be adjusted according to the differences between its clock and its neighbors' clocks (assuming node i has exchanged clock readings with its neighbors). That is, the clock at *Node* i should be set to $c_i - \sum_{j \neq i} r_{ij}(c_i - c_j)$, where $r_{ij} > 0$ is the diffusion rate, $r_{ij} = 0$ if *Node* i and *Node* j cannot directly communicate and the condition $\sum_{j \neq i} r_{ij} \leq 1$ is enforced. The above algorithm can also be formulated using matrix notation. For a group of n sensor nodes, let $\mathbf{c}^t$ be the vector of length n containing the clock readings of all the sensor nodes at time t. The synchronous diffusion algorithm adjusts the clocks of different nodes using $\mathbf{c}^{t+1} = \mathbf{R}\mathbf{c}^t$, where

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix} \tag{4.8}$$

and $r_{ii} = 1 - \sum_{j \neq i} r_{ij}$. It is shown in [47] that if the second largest eigenvalue of $\mathbf{R}$ is smaller than 1, the synchronous diffusion algorithm will converge, in the sense that all the elements in $\mathbf{c}^t$ will be equal.

The synchronous diffusion algorithm requires all the nodes to operate in an ordered manner. In order to remove this constraint, [47] proposed another algorithm, named asynchronous diffusion algorithm. In this algorithm, each node asks its neighbors about their clock readings and compute the average value. Then the average value is sent back to the neighbors so they can update their clocks. This algorithm gives a very simple averaging operation of a node over its neighbors and the averaging operations by different nodes can be carried out at different times and in any order (thus the name asynchronous). It is shown in [47] that the clocks of sensor nodes at a sensor network converges to the average value by using this asynchronous algorithm.

8. Protocols Based on Pulse Transmissions

Recently, synchronization schemes that operate exclusively at the physical layer by transmitting pulses instead of message packets have been proposed in [48] and [49]. In [48], inspired by the synchronously flashing fireflies, the time synchronization problem in sensor network is modeled using the pulse coupled oscillators (PCO). In this scheme, each node (say *Node* j) in the sensor network is associated with an increasing monotonic state function $x_j(t)$ taking values from 0 to 1. If a node is isolated, the state function $x_j(t)$ increases from 0 to 1 smoothly as a function of time and the node emits a pulse when the state function achieves the unit value ($x_j(t) = 1$). After firing a pulse, the node resets immediately its state to zero. This results in periodic emission of pulses with period T. If a node is not isolated, it can receive pulses from

other nodes. When a node receive a pulse, its state variable changes as follow

$$x_j(\tau^+) = \begin{cases} x_j(\tau) + \varepsilon, & \text{if } x_j(\tau) + \varepsilon < 1 \\ 0, & \text{otherwise} \end{cases}, \tag{4.9}$$

where τ is the time the node receive a pulse and ε is the advancement of the clock phase. This means that a node receiving a pulse either emits the pulse at the same time or shortens the waiting time for the next round of emissions. With the assumption that after a node fires a pulse, it enters a short refractory period, during which no signal can be received from other nodes (to avoid infinite feedback). It can be shown that only when the nodes emit the pulse simultaneously they will be insensitive to coupling, and therefore achieve synchronization.

In [49], a cooperative technique that constructs a sequence of pulses with equidistance zero-crossings is developed. The basic idea of this scheme is as follows. Assume there is a leader node and it emits a sequence of pulses with equidistance zero-crossings. The surrounding nodes receive this pulse sequence, and based on the locations of the observed zero-crossings, the surrounding nodes predict when the next pulse will be transmitted. Then, these nodes emit pulses at their predicted times and an aggregate pulse sequence will be generated. It is shown in [49] that although the prediction at individual node may not be perfect, under certain conditions on the pulse and in asymptotically dense networks, the zero-crossings of the aggregate waveform sequence will be at the same positions as the zero-crossings of the original waveform sequence emitted by the leader node due to spatial averaging. This aggregate pulse sequence will be heard by the nodes lying further away from the leader node and these nodes perform prediction as described before and emit their pulses at their predicted times. The procedure will be continued until all the nodes are synchronized.

Notice that the synchronization algorithms discussed in this subsection only provide a unified ticking rhythm across sensor nodes, but not the synchronization of clock time. A good analogy is a group of people clapping together to get a rhythm. However, there exist applications in which a unified rhythm is enough, e.g., in distributed beamforming and reachback channel [50]. As another variation, a joint physical- and network-layer time synchronization scheme was proposed to overcome the effects of imperfect physical layer synchronization due to the nature of common wireless channels [51].

C. Adaptive Time Synchronization

While all the above mentioned protocols in this chapter can achieve instantaneous synchronization among nodes, the timing of different nodes would drift apart as time passes; therefore, periodic re-synchronization is needed to maintain long-term synchronization. Intuitively, less frequent re-synchronization requires lesser energy but leads to a larger synchronization error, while more frequent re-synchronization leads to a smaller synchronization error but requires more energy. A natural question is what is the minimum re-synchronization frequency (or equivalently maximum re-synchronization period) that can meet the desired synchronization precision. Therefore, adaptive algorithms are necessary to dynamically determine the re-synchronization period, number of beacons to be used in each round of synchronization, synchronization accuracy, and so on. In this section, we will review three existing adaptive time synchronization algorithms proposed in the literature.

1. Rate-Adaptive Time Synchronization (RATS)

Consider the case where *Node A* sends time-stamped messages to *Node B* periodically with period τ, and *Node B* records the receiving times of the messages. Based on a number of data points $(T_i^{(A)}, T_i^{(B)})$, where $T_i^{(A)}$ and $T_i^{(B)}$ are the time-stamps made at *Node A* and *Node B*, respectively, *Node B* wants to determine the largest τ such that the synchronization error is smaller than a certain limit. The Rate-Adaptive Time Synchronization [52] is an algorithm that determines the optimal τ adaptively. Its idea can be summarized using the flow chart shown in Fig. 8. First, *Node B* calculates the optimal number of data samples for model parameters (e.g., clock offset and skew) estimation based on the current value of τ. Next, *Node B* takes the required number of data points (stored in memory) and estimates the model parameters. Then, *Node B* computes the prediction error. Finally, using the calculated prediction error, *Node B* adjusts the frequency of getting a new timing message from *Node A*: if the prediction error is larger than the upper limit threshold E_u, it means that the timing message rate is not frequent enough from *Node A*, therefore τ should be decreased; on the other hand, if the prediction error is smaller than the lower limit threshold E_l, that translates into fewer timing messages, thus τ should be increased. Multiplicative increase and decrease strategies are used to enable fast convergence and quick response to the changing environment. After getting a new data point according to the new value of τ, the above process is repeated.

2. RBS-Based Adaptive Clock Synchronization

With in the RBS setting, [53] extends the deterministic RBS protocol (discussed in Section 3) to an adaptive probabilistic synchronization algorithm, allowing trade-offs between synchronization accuracy and resource expenditure. It is based on the

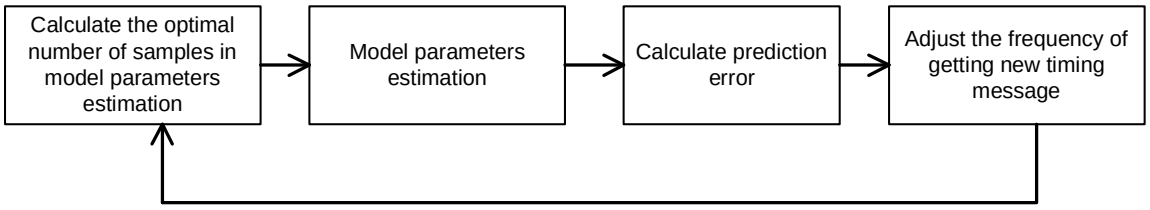

Fig. 8. Flow chart of RATS run at the node receiving time-stamped messages from another node.

observation if the relative clock skew error between two nodes ε is a Gaussian RV with zero mean and variance σ^2, then the probability of error free synchronization with N broadcasting messages is given by

$$Pr(|\varepsilon| < \varepsilon_{max}) = 2erf\left(\frac{\sqrt{N}\varepsilon_{max}}{\sigma}\right), \tag{4.10}$$

where ε_{max} stands for the maximum specified (allowable) clock offset for communications, and $erf(x) \triangleq (1/2\pi) \cdot \int_0^x \exp(-t^2/2)dt$. From the above equation, it is clear that the performance criterion is a probabilistic measure since there is always a possibility that the clock offset is greater than some limit ε_{max}. However, one can reduce this probability to an arbitrarily small value by increasing N, the number of broadcasting messages in one round of RBS.

After application of RBS, we can bound the clock skew error with certain probability. However, since clocks from different nodes would drift apart as time passes, we need to re-apply RBS periodically. Reference [53] proposes a formula to determine the maximum time between re-synchronization $\tau_{\max}$ as

$$\tau_{\max} = \frac{\gamma_{max} - \varepsilon_{max}}{\rho} - d_{max}, \tag{4.11}$$

where γ_{max} denotes the maximum allowable clock skew at any time, ρ denotes the maximum drift of clock, and d_{max} is the maximum delay of time-stamp exchanges in RBS. With different synchronization precision requirements (specified by γ_{max}), one

can determine the required re-synchronization period $\tau_{\max}$.

3. Adaptive Multi-Hop Time Synchronization

In this dissertation, we propose the Adaptive Multi-Hop Time Synchronization (AMTS) protocol, which is based on a similar system model as in TPSN and employees a number of novel features as well [28]. It consists of three functional phases: *network level discovery phase*, *synchronization phase*, and *network evaluation phase*, and a number of network parameters such as latency factor, average number of hops, and re-synchronization period to optimize the synchronization protocol. Relative to TPSN, AMTS assumes the additional *network evaluation phase*, while the functions of the other two phases are similar to the ones encountered in TPSN.

Robustness to high-latencies and network delays is ensured based on the clock estimators presented in this dissertation, and therefore AMTS fits well for sensor network applications having large delays in timing message exchanges such as underwater acoustic sensor networks [54]. Besides, AMTS adapts the joint clock offset and skew estimators to increase the re-synchronization period.

As TPSN, generating a hierarchical structure in the network, the level discovery phase, is the first step of AMTS. In this phase, every single node in the network will be assigned a level and is ready for synchronization. The second step of AMTS, called the time synchronization phase, consists in pairwise synchronizations between adjacent nodes until every node in the network is synchronized to the reference. In the synchronization phase, AMTS estimates not only the current clock offset but also the clock frequency (skew) to guarantee long term reliability of synchronization while TPSN only estimates the clock offset. Hence, it requires far less frequent re-synchronization. Finally, the reference node investigates the current status of network traffic in order to optimize the re-synchronization period and the number of beacons in

terms of energy efficiency. Besides it selects the synchronization mode between *always on* (**AO**) (always maintain network-wide synchronization) and *sensor initiated* (**SI**) (synchronize only when it needs to) based on the network status. This step stands for the network evaluation phase, and its goal is to minimize the number of message exchanges for synchronization in a given time, i.e., it aims to minimize total energy consumption for synchronization. AMTS periodically repeats the synchronization and network evaluation phases to minimize the total energy consumption with respect to the current network status. Chapter VII describes these procedures in detail.

CHAPTER V

CLOCK OFFSET AND SKEW ESTIMATION USING TWO-WAY MESSAGE EXCHANGES

A. Motivations

As introduced in Chapter IV, a number of synchronization protocols have been reported for synchronizing the nodes of WSNs. These protocols are subject to their own benefits as well as limitations. For protocols which correct only the clock offset (such as TPSN [14]), synchronization has to be done frequently at regular intervals to prevent the clock skew drift the two clocks too far apart, hence utilizing more energy resources. For example, re-synchronization must be performed every a few minutes in TPSN for applications using the *MICA* platform [20], [21]. On the other hand, protocols which correct both the clock offset and skew (such as RBS [27] and FTSP [22]) assume simultaneous reception of reference broadcasts, which is not applicable in some cases, e.g., in underwater acoustic sensor networks [54]. In [54], it has been asserted that for this type of sensor networks, there are large variations in packet delays between nodes resulting in significant synchronization error. Thus, an adequate solution would be the TPSN protocol, provided that the clock skew can also be estimated along with the clock offset.

This chapter analyzes the clock synchronization protocols relying on two-way message exchanges between the nodes, a set-up similar to TPSN. A thorough analysis of two-way message exchange between two nodes under the symmetric exponential noise model is carried out by [31]. Assuming that the exponential noise parameter α and the fixed portion of the delays (d) are known, [31] has argued that the MLE of clock time offset (θ_{o}) does not exist because the likelihood function does not possess

a unique maximum with respect to θ_o. Recently, it has been shown in [32] that the MLE of θ_o exists when d is unknown, and it coincides to the estimator proposed in [33].

The contributions of this study are as follows. First, we analyze and derive the maximum likelihood estimators (MLEs) and corresponding Cramer-Rao lower bounds (CRBs) for the conventional clock offset model as used in [14], assuming Gaussian and exponential models for the noise, respectively. Second, we derive the joint MLE and corresponding CRB using a more realistic linear clock skew model assuming Gaussian random delays. Third, novel and practical clock skew estimators, which do not require to know the fixed portion of delays, are proposed. The introduction of a clock skew correction mechanism prolongs the re-synchronization period significantly, and therefore far less power resources will be required in the synchronization process. In fact, the proposed clock synchronization mechanism can be directly applied to the conventional protocols using simple and low complexity modifications, a feature which is strongly demanding for WSNs consisting of cheap and small nodes.

The rest of the chapter is organized as follows. In Section B, the MLEs of clock offset are analyzed and the corresponding CRB are derived for exponential and Gaussian random delays, respectively. Section C presents the clock skew model adopted in this Chapter and derives the corresponding joint ML clock offset and skew estimator for the Gaussian random delay model. Section D proposes practical and robust clock skew and offset estimators for both exponential and Gaussian random delays, respectively. In Section E, various computer simulation results are provided for performance evaluations, and finally Section F concludes the study.

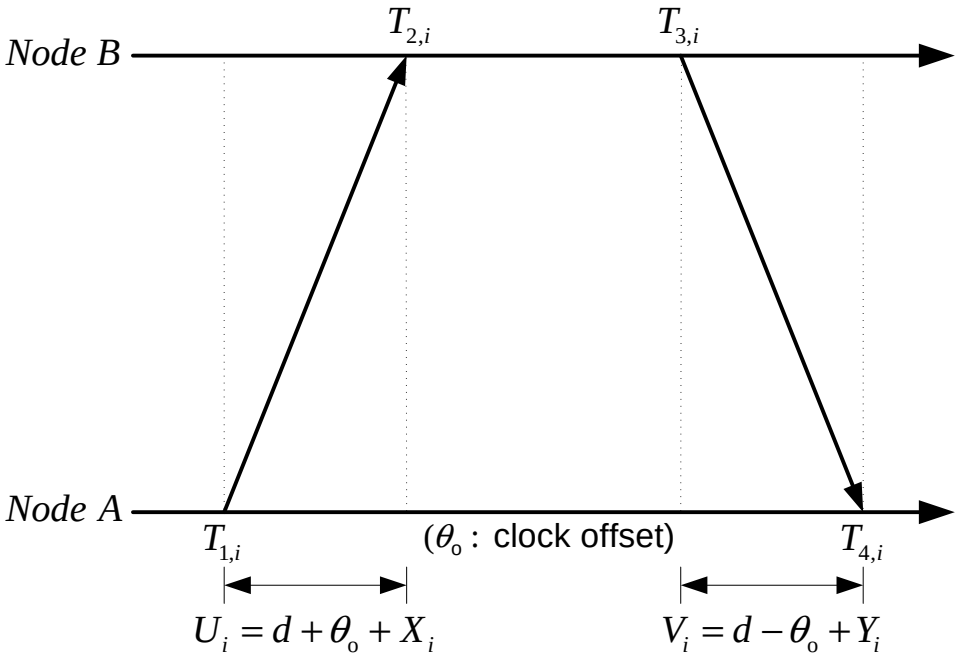

Fig. 9. Two-way timing message exchange model between master-slave nodes assuming only clock offset.

B. Maximum Likelihood Clock Offset Estimation

Assuming no clock skew at this stage, we compute the MLE and CRB for the clock offset using the two-way timing message exchange model. This scenario is depicted in Fig. 9, where *Node A* sends its time reading $T_{1,i}$ to *Node B*, which records its time of arrival $T_{2,i}$ according to its own timescale. A similar timing message exchange is performed from *Node B* to *Node A*, as shown in Fig. 9.

Thus far, several probability density function (PDF) models have been proposed for modeling random queuing delays, the most widely deployed of which are Gamma, exponential and Weibull PDFs [29], [30]. As explained in [31], a single-server M/M/1 queue can fittingly represent the cumulative link delay for point-to-point Hypothetical Reference Connection, where the random delays are independently modeled as exponential random variables (RVs). The reason for adopting Gaussian PDF is due to

the Central Limit Theorem, which asserts that the PDF of the sum of a large number of independent and identically distributed (iid) RVs approaches that of a Gaussian RV. This model will be appropriate if the delays are thought to be the addition of numerous independent random processes. The Gaussian distribution for the phase offset errors is reported by a few authors, such as [27], based on laboratory tests.

The ith up and down link delay observations corresponding to the ith timing message exchange are given by $U_i \triangleq T_{2,i} - T_{1,i} = d + \theta_{\mathrm{o}} + X_i$ and $V_i \triangleq T_{4,i} - T_{3,i} = d - \theta_{\mathrm{o}} + Y_i$, respectively (using similar notations as in [31]), and are graphically represented in Fig. 9. The fixed value θ_{o} denotes the clock offset between two nodes, X_i and Y_i denote the variable portions of delays which are assumed to be either exponentially distributed RVs with means λ_1 and λ_2 or normal distributed RVs with mean μ and variance σ^2, respectively.

1. Exponential Delay Model

a. Cramer-Rao Lower Bound

It was proven in [32] that the MLE of θ_{o} exists when d is unknown and exhibits the same form as the estimator proposed in [33], which is given by

$$\hat{\theta}_{\mathrm{o}} = \frac{\min\limits_{1\le i\le N} U_i - \min\limits_{1\le i\le N} V_i}{2}, \tag{5.1}$$

where N stands for the number of observations of delay measurements. For simpler notations and further analysis, let $\left\{U_{(i)}\right\}_{i=1}^{N}$ and $\left\{V_{(i)}\right\}_{i=1}^{N}$ denote the order statistics of the sequences of delay observations $\{U_i\}_{i=1}^{N}$ and $\{V_i\}_{i=1}^{N}$, respectively. Then (5.1) can be rewritten as

$$\hat{\theta}_{\mathrm{o}} = \frac{U_{(1)} - V_{(1)}}{2} = \theta_{\mathrm{o}} + \frac{X_{(1)} - Y_{(1)}}{2},$$

where $X_{(1)}$ and $Y_{(1)}$ denote the corresponding order statistics of $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$, respectively. Let $Z \triangleq X_{(1)} - Y_{(1)}$, then from the result in Appendix A, the PDF of Z is given by

$$f_Z(z) = \begin{cases} \frac{N}{(\lambda_1+\lambda_2)} e^{-\frac{N}{\lambda_1}z} & z > 0 \\ \frac{N}{(\lambda_1+\lambda_2)} e^{\frac{N}{\lambda_2}z} & z < 0 \end{cases}. \tag{5.2}$$

Let $W \triangleq U_{(1)} - V_{(1)}$, then the PDF of W as a function of θ_o is given by

$$f_W(w;\theta_o) = \begin{cases} \frac{N}{(\lambda_1+\lambda_2)} e^{-\frac{N}{\lambda_1}(w-2\theta_o)} & w > 2\theta_o \\ \frac{N}{(\lambda_1+\lambda_2)} e^{\frac{N}{\lambda_2}(w-2\theta_o)} & w < 2\theta_o \end{cases}. \tag{5.3}$$

Note that the estimate $\hat{\theta}_o$ will be biased when uplink and downlink delays are asymmetrically distributed, i.e., $\lambda_1 \neq \lambda_2$. Thus, to derive the CRB for the estimator, the delays are assumed to be symmetric, which yields $\lambda_1 = \lambda_2 = \alpha$. Now (5.3) can be rewritten as

$$f_W(w;\theta_o) = \frac{N}{2\alpha} e^{-\frac{N}{\alpha}|w-2\theta_o|}.$$

Differentiating the logarithm of (5.3) with respect to θ_o gives

$$\frac{\partial \ln f_W(w;\theta_o)}{\partial \theta_o} = \begin{cases} \frac{2N}{\alpha} & w > 2\theta_o \\ -\frac{2N}{\alpha} & w < 2\theta_o \end{cases}, \tag{5.4}$$

where the regularity condition of the CRB [35, p. 30] holds since (5.4) is finite and the expected value of (5.4) is 0. Calculating the expected value of the square of (5.4) gives

$$E\left[\left(\frac{\partial \ln f_W(w;\theta_o)}{\partial \theta_o}\right)^2\right] = \frac{4N^2}{\alpha^2}.$$

Therefore, the CRB of clock offset, $\hat{\theta}_o$, is given by

$$\mathrm{var}(\hat{\theta}_o) \geq E\left[\left(\frac{\partial \ln f_W(w;\theta_o)}{\partial \theta_o}\right)^2\right]^{-1} = \frac{\alpha^2}{4N^2}. \tag{5.5}$$

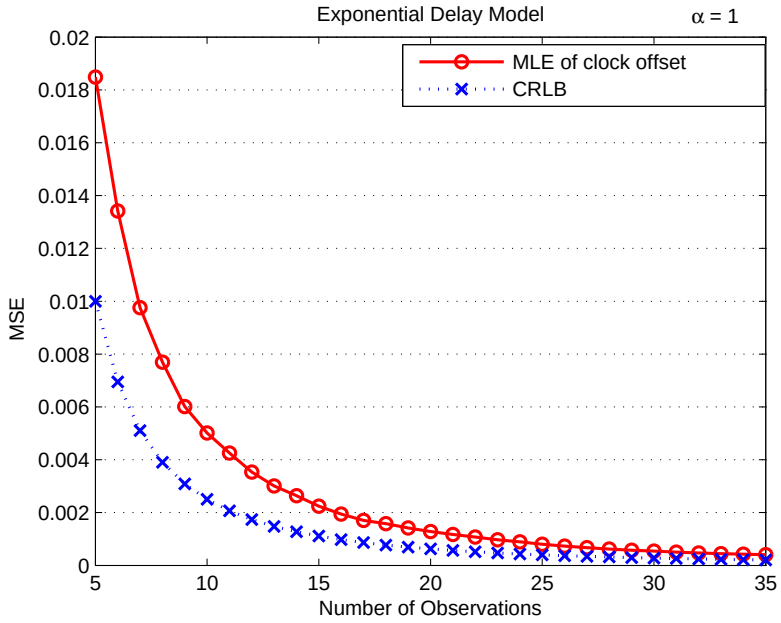

Fig. 10. CRB and MSE of the MLE of clock offset for the exponential delay model ($\alpha = 1$).

Fig. 10 shows the simulation results corresponding to the variance and CRB of the MLE when α is 1. It can be seen that the variance of estimate goes to zero as N increases (quadratic dependence), and is proportional to α^2.

2. Gaussian Delay Model

a. Maximum Likelihood Estimator

Assuming the set of delay observations $\{X_i\}_{i=1}^{N}$ and $\{Y_i\}_{i=1}^{N}$ are independently and normally distributed with the same mean μ and variance σ^2, the likelihood function based on the observations $\{X_i\}_{i=1}^{N}$ and $\{Y_i\}_{i=1}^{N}$ is given by

$$L\left(\theta_o, \mu, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(U_i - d - \theta_o - \mu)^2 + \sum_{i=1}^{N}(V_i - d + \theta_o - \mu)^2\right]}.$$

Differentiating the log-likelihood function gives

$$\begin{aligned}\frac{\partial \ln L(\theta_o)}{\partial \theta_o} &= -\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(2\theta_o - 2(U_i - d - \mu)) + \sum_{i=1}^{N}(2\theta_o + 2(V_i - d - \mu))\right] \\ &= -\frac{1}{\sigma^2}\left[\sum_{i=1}^{N}(2\theta_o - (U_i - V_i))\right]. \qquad (5.6)\end{aligned}$$

Hence the MLE of clock offset is given by

$$\hat{\theta}_o = \arg\max_{\theta_o}[\ln L(\theta_o)] = \frac{\sum_{i=1}^{N}(U_i - V_i)}{2N} = \frac{\overline{U} - \overline{V}}{2}. \qquad (5.7)$$

Consequently, the MLE of clock offset can be obtained by finding the means of observations $\{U_i\}_{i=1}^{N}$ and $\{V_i\}_{i=1}^{N}$.

b. Cramer-Rao Lower Bound

The regularity condition [35, p. 30] holds for the given estimate since the expected value of (5.6) is 0. Thus, the CRB for the MLE can be obtained by differentiating (5.6) w.r.t. θ_o, which gives

$$\frac{\partial^2 \ln L(\theta_o)}{\partial \theta_o^2} = -\frac{2N}{\sigma^2}.$$

Hence the CRB for the MLE is given by

$$\mathrm{var}(\hat{\theta}_o) \geq -E\left[\frac{\partial^2 \ln L(\theta_o)}{\partial \theta_o^2}\right]^{-1} = \frac{\sigma^2}{2N}. \qquad (5.8)$$

Fig. 11 shows the result of the computer simulation when σ is 1. It can be seen that the variance of estimate is proportional to σ^2 and inversely proportional to N.

In Fig. 12, the variances of both MLEs are compared in exponential and normal random delay channels, respectively. It can be seen that the performance of the ML clock offset estimator is strongly dependent on the type of random delay models.

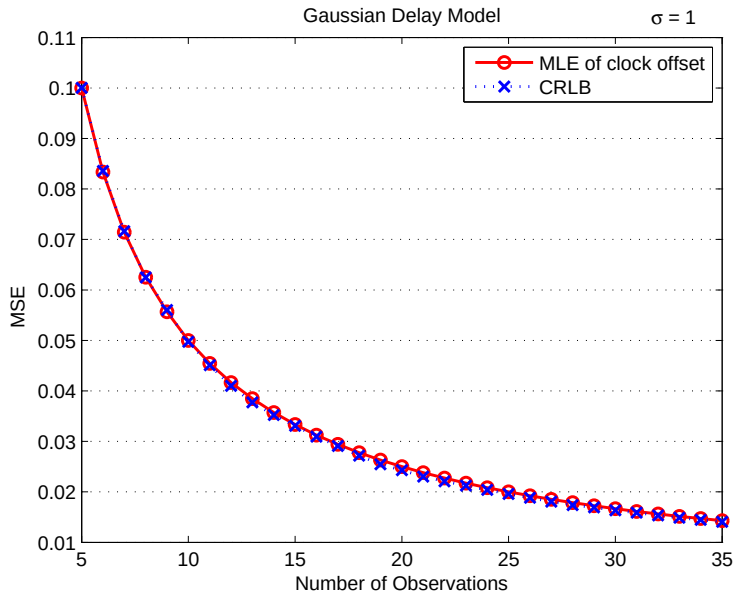

Fig. 11. CRB and MSE of the MLE of clock offset for the Gaussian delay model ($\sigma = 1$).

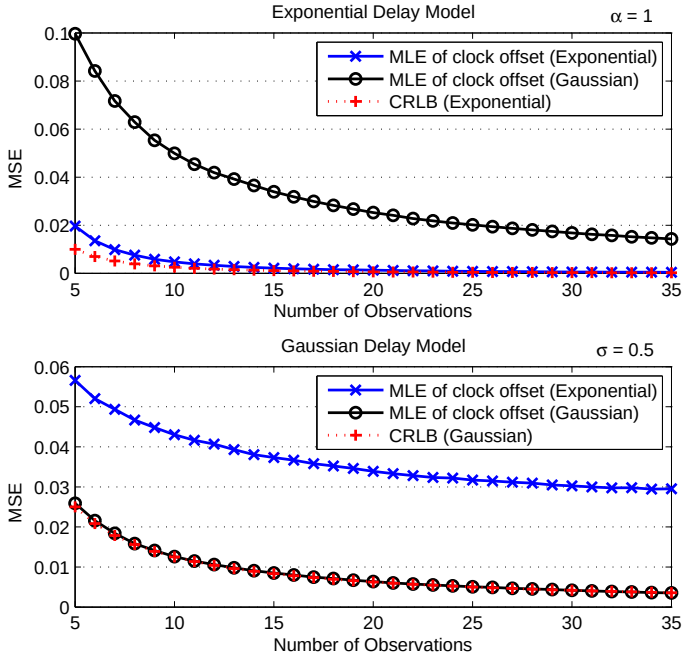

Fig. 12. MSEs of both MLEs of clock offset for exponential and Gaussian delays ($\alpha = 1$ and $\sigma = 0.5$).

C. Maximum Likelihood Clock Skew Estimation

Since every oscillator has its unique clock frequency, the clock offset between two nodes generally keeps increasing. Therefore, a fixed value model for clock time difference as in Fig. 9 is not sufficient for practical situations. Hence, estimating the difference of clock frequencies between two nodes (i.e., clock skew) increases synchronization accuracy and guarantees long-term reliability. In this section, we derive the joint MLE for clock offset and skew based on the two-way timing message exchange model with Gaussian delays.

1. Joint Maximum Likelihood Estimation of Clock Offset and Skew

The theory applied thus far for finding the MLE and CRB for the clock offset (assuming no clock skew) can be extended to find the joint MLE and CRB for a more general clock model. Fig. 13 shows the effect of clock offset ($\theta_{\rm o}^{(\rm AP)}$) and skew ($\theta_{\rm s}^{(\rm AP)}$) on timing message exchanges between *Node A* and *Node P* (using the similar notations as in [31]). Here, the time stamps in the ith message exchange $T_{1,i}^{(\rm A)}$ and $T_{4,i}^{(\rm A)}$ are measured by the local clock of *Node A*, and $T_{2,i}^{(\rm P)}$ and $T_{3,i}^{(\rm P)}$ are measured by the local clock of *Node P*, respectively. *Node A* transmits a synchronization packet, containing the level and ID of *Node A* and the value of time stamp $T_{1,i}^{(\rm A)}$, to *Node P*. *Node P* receives it at $T_{2,i}^{(\rm P)}$ and transmits an acknowledgement packet to *Node A* at $T_{3,i}^{(\rm P)}$. This packet contains the level and ID of *Node P* and the value of time stamps $T_{1,i}^{(\rm A)}$, $T_{2,i}^{(\rm P)}$, and $T_{3,i}^{(\rm P)}$. Finally, *Node A* receives the packet at $T_{4,i}^{(\rm A)}$.

Note that the reference time $T_{1,1}^{(\rm A)}$ can be set to be zero without loss of generality. Then, the time stamp at *Node P* in the ith uplink message $T_{2,i}^{(\rm B)}$ is given by

$$\begin{aligned} T_{2,i}^{(\rm P)} &= T_{1,i}^{(\rm A)} + \theta_{\rm o}^{(\rm AP)} + \theta_{\rm s}^{(\rm AP)}(T_{1,i}^{(\rm A)} + d + X_i^{(\rm AP)}) + d + X_i^{(\rm AP)} \\ &= (1+\theta_{\rm s}^{(\rm AP)})(T_{1,i}^{(\rm A)} + d + X_i^{(\rm AP)}) + \theta_{\rm o}^{(\rm AP)}, \end{aligned} \tag{5.9}$$

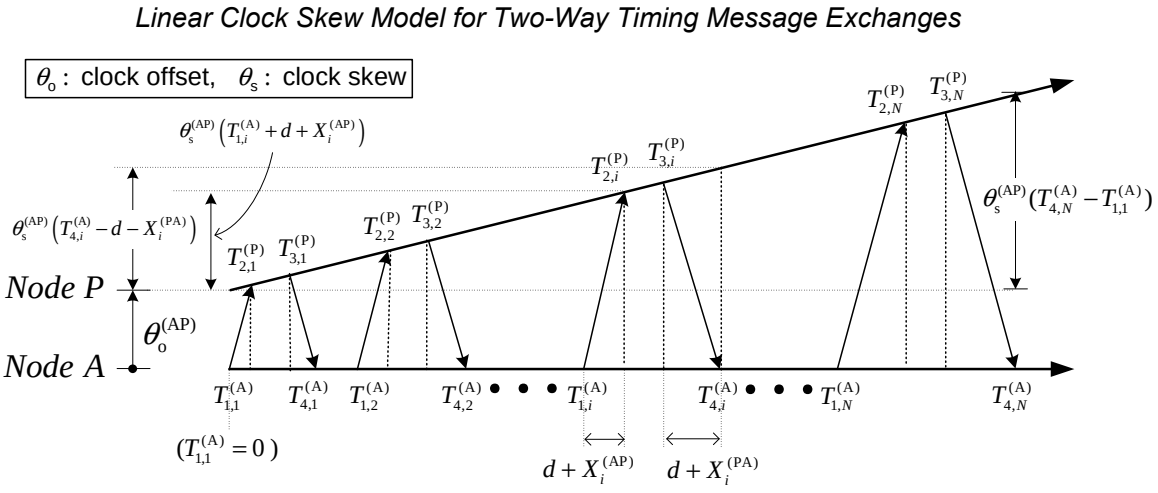

Fig. 13. Two-way timing message exchange model assuming clock offset and skew.

where the term $\theta_{\rm s}^{(\rm AP)}(T_{1,i}^{(\rm A)} + d + X_i^{(\rm AP)})$ is due to the effect of clock skew. Similarly, the time stamp at *Node P* in the ith downlink message $T_{3,i}^{(\rm P)}$ takes the equations

$$\begin{aligned} T_{3,i}^{(\rm P)} &= T_{4,i}^{(\rm A)} + \theta_{\rm o}^{(\rm AP)} + \theta_{\rm s}^{(\rm AP)}(T_{4,i}^{(\rm A)} - d - X_i^{(\rm PA)}) - d - X_i^{(\rm PA)} \\ &= (1 + \theta_{\rm s}^{(\rm AP)})(T_{4,i}^{(\rm A)} - d - X_i^{(\rm PA)}) + \theta_{\rm o}^{(\rm AP)}, \end{aligned} \tag{5.10}$$

where the term $\theta_{\rm s}^{(\rm AP)}(T_{4,i}^{(\rm A)} - d - X_i^{(\rm PA)})$ is again due to the effect of clock skew. For an easier illustration, we introduce the simplified notations $\theta_{\rm s} \triangleq \theta_{\rm s}^{(\rm AP)}$, $\theta_{\rm o} \triangleq \theta_{\rm o}^{(\rm AP)}$, $T_{1,i} \triangleq T_{1,i}^{(\rm A)}$, $T_{4,i} \triangleq T_{4,i}^{(\rm A)}$, $T_{2,i} \triangleq T_{2,i}^{(\rm P)}$, $T_{3,i} \triangleq T_{3,i}^{(\rm P)}$, $X_i \triangleq X_i^{(\rm AP)}$, and $Y_i \triangleq X_i^{(\rm PA)}$ in this subsection, respectively.

Assuming $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$ are zero mean independent Gaussian distributed RVs with variance σ^2, then the joint PDF of $\mathbf{X} \triangleq \{X_i\}_{i=1}^N$ and $\mathbf{Y} \triangleq \{Y_i\}_{i=1}^N$ is given by

$$f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = \left(2\pi\sigma^2\right)^{-\frac{N}{2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left[\left(\frac{\theta_{\rm o}-T_{2,i}+(T_{1,i}+d)(1+\theta_{\rm s})}{1+\theta_{\rm s}}\right)^2+\left(\frac{\theta_{\rm o}-T_{3,i}+(T_{4,i}-d)(1+\theta_{\rm s})}{1+\theta_{\rm s}}\right)^2\right]}.$$

Further assuming that the fixed portion of delay d is known and $\theta_{\rm s}' \triangleq 1/(1+\theta_{\rm s})$,

then the likelihood function for $(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2)$, based on observations $\{T_{1,i}\}_{i=1}^{N}$, $\{T_{2,i}\}_{i=1}^{N}$, $\{T_{3,i}\}_{i=1}^{N}$, and $\{T_{4,i}\}_{i=1}^{N}$, is given by

$$L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-\frac{N}{2}} e^{-\frac{1}{2\sigma^2}\sum\limits_{i=1}^{N}\left\{[\theta'_{\rm s}(\theta_{\rm o}-T_{2,i})+(T_{1,i}+d)]^2+[\theta'_{\rm s}(\theta_{\rm o}-T_{3,i})+(T_{4,i}-d)]^2\right\}}.$$

Differentiating the log-likelihood function with respect to $\theta_{\rm o}$ gives

$$\frac{\partial \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta_{\rm o}} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}\Big[\theta'^2_{\rm s}\left(2\theta_{\rm o} - T_{2,i} - T_{3,i}\right) + \theta'_{\rm s}\left(T_{1,i} + T_{4,i}\right)\Big]. \tag{5.11}$$

Hence, in the given clock skew model, the joint MLE of clock offset $\hat{\theta}_{\rm o}$ can be expressed as

$$\begin{aligned}\hat{\theta}_{\rm o} &= \frac{\sum_{i=1}^{N}\Big[\hat{\theta}'_{\rm s}\left(T_{2,i}+T_{3,i}\right) - \left(T_{1,i}+T_{4,i}\right)\Big]}{2N\hat{\theta}'_{\rm s}} \\ &= \frac{\overline{U}-\overline{V}}{2} - \hat{\theta}_{\rm s}\frac{\overline{T_4}+\overline{T_1}}{2},\end{aligned} \tag{5.12}$$

where $\overline{T_i}$ stands for the average value of T_i ($\overline{T_i} \triangleq \sum_{j=1}^{N} T_{i,j}/N$).

Note that the clock offset estimate (5.12) in the case of the clock skew model with Gaussian random delays presents an additional term which depends on $\hat{\theta}_{\rm s}$, and this expression reduces to (5.7) when $\hat{\theta}_{\rm s}$ is zero. Similarly, differentiating the log-likelihood function with respect to $\theta'_{\rm s}$ gives

$$\begin{aligned}\frac{\partial \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta'_{\rm s}} &= -\frac{1}{\sigma^2}\Bigg\{\sum_{i=1}^{N}\theta'_{\rm s}\left[(T_{2,i}-\theta_{\rm o})^2 + (T_{3,i}-\theta_{\rm o})^2\right] \\ &\qquad -\sum_{i=1}^{N}\left[(T_{1,i}+d)(T_{2,i}-\theta_{\rm o}) + (T_{4,i}-d)(T_{3,i}-\theta_{\rm o})\right]\Bigg\}.\end{aligned} \tag{5.13}$$

Thus, the estimate $\hat{\theta}'_{\rm s}$ maximizing the log-likelihood function is given by

$$\hat{\theta}'_{\rm s} = \frac{\sum_{i=1}^{N}\Big[(T_{1,i}+d)(T_{2,i}-\hat{\theta}_{\rm o}) + (T_{4,i}-d)(T_{3,i}-\hat{\theta}_{\rm o})\Big]}{\sum_{i=1}^{N}\Big[(T_{2,i}-\hat{\theta}_{\rm o})^2 + (T_{3,i}-\hat{\theta}_{\rm o})^2\Big]}.$$

Hence, the joint MLE of clock skew $\hat{\theta}_{\rm s}$ is given by

$$\hat{\theta}_{\rm s} = \frac{\sum_{i=1}^{N}\left[(T_{2,i}-\hat{\theta}_{\rm o})^2+(T_{3,i}-\hat{\theta}_{\rm o})^2\right]}{\sum_{i=1}^{N}\left[(T_{1,i}+d)(T_{2,i}-\hat{\theta}_{\rm o})+(T_{4,i}-d)(T_{3,i}-\hat{\theta}_{\rm o})\right]}-1. \tag{5.14}$$

In the sequel, the joint MLE of $\theta_{\rm o}$ and $\theta_{\rm s}$ can be obtained by plugging the expression of $\hat{\theta}_{\rm o}$ (5.12) into that of $\hat{\theta}_{\rm s}$ (5.14). From the result in Appendix B, the joint MLE of $\theta_{\rm o}$ and $\theta_{\rm s}$ can be expressed as

$$\hat{\theta}_{\rm o}^{GML} = \frac{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\sum_{i=1}^{N}(T_{2,i}^2+T_{3,i}^2)-\sum_{i=1}^{N}(T_{2,i}+T_{3,i})Q}{\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\sum_{i=1}^{N}(T_{1,i}+T_{4,i})-2NQ}, \tag{5.15}$$

$$\begin{aligned}\hat{\theta}_{\rm s}^{GML} = {} & \frac{-2N\left[\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\sum_{i=1}^{N}(T_{2,i}^2+T_{3,i}^2)-Q\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\right]}{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\left[\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\sum_{i=1}^{N}(T_{1,i}+T_{4,i})-2NQ\right]} \\ & +\frac{\sum_{i=1}^{N}(T_{2,i}+T_{3,i})}{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})}-1,\end{aligned} \tag{5.16}$$

where $Q \triangleq \sum_{i=1}^{N}(T_{1,i}T_{2,i}+T_{3,i}T_{4,i}+(T_{2,i}-T_{3,i})d)$. Note that the joint MLE depends on the value of the fixed portion of delays d, which is assumed to be known in this section. Although estimating d is an achievable task, we do not consider d as another unknown (nuisance) parameter due to the inherent highly nonlinear and complex operations required for estimating d.

2. Cramer-Rao Lower Bound for the Joint MLE

The CRB for the vector parameter $\boldsymbol{\theta} = [\theta_{\rm o}, \theta_{\rm s}]^T$ can be derived from the 2×2 Fisher information matrix $\mathbf{I}(\boldsymbol{\theta})$ by taking its inverse. From (5.11) and (5.13), the 2nd order

derivatives of the log-likelihood function with respect to $\theta_{\rm o}$ and $\theta'_{\rm s}$ are found as

$$\begin{aligned}
\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta_{\rm o}^2} &= -\frac{2N{\theta'_{\rm s}}^2}{\sigma^2}, \\
\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial {\theta'_{\rm s}}^2} &= -\frac{1}{\sigma^2}\sum_{i=1}^{N}\left[(T_{2,i}-\theta_{\rm o})^2 + (T_{3,i}-\theta_{\rm o})^2\right], \\
\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta_{\rm o}\theta'_{\rm s}} &= -\frac{1}{\sigma^2}\sum_{i=1}^{N}(2\theta'_{\rm s}\theta_{\rm o} - \theta'_{\rm s}T_{2,i} + T_{1,i} - \theta'_{\rm s}T_{3,i} - T_{4,i}).
\end{aligned}$$

Taking the negative expectations yields

$$\begin{aligned}
-E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta_{\rm o}^2}\right] &= \frac{2N{\theta'_{\rm s}}^2}{\sigma^2}, \\
-E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial {\theta'_{\rm s}}^2}\right] &= \frac{1}{\sigma^2}\sum_{i=1}^{N} E_{X_i,Y_i}\left[\frac{(X_i + T_{1,i} + d)^2 + (Y_i - T_{4,i} + d)^2}{{\theta'_{\rm s}}^2}\right] \\
&\overset{(a)}{=} \frac{\sum_{i=1}^{N}\left((T_{1,i}+d)^2 + (T_{4,i}-d)^2 + 2\sigma^2\right)}{\sigma^2{\theta'_{\rm s}}^2}, \\
-E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o}, \theta'_{\rm s}, \sigma^2\right)}{\partial \theta_{\rm o}\theta'_{\rm s}}\right] &= -\frac{1}{\sigma^2}\sum_{i=1}^{N} E_{X_i,Y_i}\left[2\theta'_{\rm s}\left(2\theta_{\rm o} - T_{2,i} - T_{3,i}\right) + T_{1,i} + T_{4,i}\right] \\
&\overset{(b)}{=} \frac{N}{\sigma^2}\left(\overline{T_1} + \overline{T_4}\right),
\end{aligned}$$

where (a) and (b) are due to $X_i = \theta'_{\rm s}(T_{2,i} - \theta_{\rm o}) - (T_{1,i} + d)$ and $Y_i = \theta'_{\rm s}(\theta_{\rm o} - T_{3,i}) + (T_{4,i} - d)$. Therefore, the Fisher information matrix becomes

$$\begin{aligned}
\mathbf{I}\left(\boldsymbol{\theta}\right) &= \begin{bmatrix} -E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o},\theta'_{\rm s},\sigma^2\right)}{\partial \theta_{\rm o}^2}\right] & -E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o},\theta'_{\rm s},\sigma^2\right)}{\partial \theta_{\rm o}\theta'_{\rm s}}\right] \\ -E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o},\theta'_{\rm s},\sigma^2\right)}{\partial \theta'_{\rm s}\theta_{\rm o}}\right] & -E\left[\frac{\partial^2 \ln L\left(\theta_{\rm o},\theta'_{\rm s},\sigma^2\right)}{\partial {\theta'_{\rm s}}^2}\right] \end{bmatrix}, \\
&= \frac{1}{\sigma^2}\begin{bmatrix} 2N{\theta'_{\rm s}}^2 & N\left(\overline{T_1} + \overline{T_4}\right) \\ N\left(\overline{T_1} + \overline{T_4}\right) & \frac{1}{{\theta'_{\rm s}}^2}\sum_{i=1}^{N}\left[(T_{1,i}+d)^2 + (T_{4,i}-d)^2 + 2\sigma^2\right] \end{bmatrix}. \qquad (5.17)
\end{aligned}$$

From [35, p. 40], the CRB can be obtained by taking the inverse of the $[i,i]$th element of the Fisher information matrix (i.e., $\mathrm{var}(\hat{\theta}_i) \geq [\mathbf{I}^{-1}(\boldsymbol{\theta})]_{ii}$), and the inverse $\mathbf{I}^{-1}(\boldsymbol{\theta})$ is

given by

$$\mathbf{I}^{-1}(\boldsymbol{\theta}) = \sigma^2 \begin{bmatrix} \frac{V}{\theta_{\rm s}'^2 N\left[2V - N\left(\overline{T_1}+\overline{T_4}\right)^2\right]} & \frac{-\left(\overline{T_1}+\overline{T_4}\right)}{2V - N\left(\overline{T_1}+\overline{T_4}\right)^2} \\ \frac{-\left(\overline{T_1}+\overline{T_4}\right)}{2V - N\left(\overline{T_1}+\overline{T_4}\right)^2} & \frac{2\theta_{\rm s}'^2}{2V - N\left(\overline{T_1}+\overline{T_4}\right)^2} \end{bmatrix}, \tag{5.18}$$

where $V = \sum_{i=1}^{N}\left[(T_{1,i}+d)^2 + (T_{4,i}-d)^2 + 2\sigma^2\right]$. Consequently, from the result in [35, p. 37], the CRBs of clock offset and skew for the Gaussian delay model are given respectively by

$$\begin{aligned} \mathrm{var}(\hat{\theta}_{\rm o}^{GML}) &\geq \frac{\sigma^2(1+\theta_{\rm s})^2 V}{N\left[2V - N\left(\overline{T_1}+\overline{T_4}\right)^2\right]}, & (5.19) \\ \mathrm{var}(\hat{\theta}_{\rm s}^{GML}) &\geq \left(\frac{\partial\theta_{\rm s}}{\partial\theta_{\rm s}'}\right)^2 \cdot \frac{2\sigma^2\theta_{\rm s}'^2}{2V - N\left(\overline{T_1}+\overline{T_4}\right)^2} \\ &= \frac{2\sigma^2\left(1+\theta_{\rm s}\right)^2}{2V - N\left(\overline{T_1}+\overline{T_4}\right)^2}. & (5.20) \end{aligned}$$

D. Proposed Clock Skew Estimators

The joint MLE of clock offset and skew for Gaussian delays has been derived in the previous section. However, for exponentially distributed delays, the joint PDF does not possess local maxima with respect to either $\theta_{\rm o}$ or $\theta_{\rm s}'$, and it assumes the highly complex expression

$$\begin{aligned} f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) &= (\alpha)^{-2N} e^{-\frac{1}{\alpha}\sum_{i=1}^{N}(\theta_{\rm s}'(T_{2,i}-T_{3,i})+T_{4,i}-T_{1,i}-2d)} \times \\ &\quad \prod_{i=1}^{N} F\left[\theta_{\rm s}'(T_{2,i}-\theta_{\rm o}) - (T_{1,i}+d) \geq 0, \theta_{\rm s}'(\theta_{\rm o}-T_{3,i}) + (T_{4,i}-d) \geq 0\right], \end{aligned}$$

where $F(\cdot)$ stands for the indicator function (i.e., $F(\cdot)$ is 1 whenever its inner condition holds, otherwise being equal to 0). Thus, an alternative estimator is required for the exponential skew model. Besides, even for the Gaussian delay model, finding the joint MLE of clock skew requires some computations as in (5.16) and the fixed portion of

delays d must be known (or estimated), which might not be applicable for wireless sensor networks consisting of low-end terminals. In practice, it requires an additional estimation procedure, which might deteriorate the robustness of the joint MLE. For these reasons, we propose simple and robust clock skew estimators for the exponential and Gaussian delay models, respectively, which do not require prior knowledge of d.

Since the clock difference between two wireless terminals is monotonically increasing (or temporary decreasing then increasing) based on the linear clock skew model adopted in this chapter, the clock difference will be maximized between the first and last time stamps. From this intuition, novel and practical clock skew estimators can be developed by using the first and last observations of timing message exchanges. In this regard, we propose an ML-Like Estimator (MLLE) that maximizes the likelihood function obtained based on a reduced subset of observations (the first and last timing stamps).

From (5.9), subtracting $T_{2,1}$ from $T_{2,N}$ leads to

$$T_{2,N} - T_{2,1} = T_{1,N} - T_{1,1} + X_N - X_1 + \theta_s \left(T_{1,N} - T_{1,1} + X_N - X_1\right). \tag{5.21}$$

Similarly from (5.10), subtracting $T_{4,1}$ from $T_{4,N}$ yields

$$T_{4,N} - T_{4,1} = T_{3,N} - T_{3,1} + Y_N - Y_1 - \theta_s \left(T_{4,N} - T_{4,1} - (Y_N - Y_1)\right). \tag{5.22}$$

Define the differences of the first and last time stamps as $D_{(1)} \triangleq \sum_{i=2}^{N} D_{1,i} = T_{1,N} - T_{1,1}$, $D_{(2)} \triangleq \sum_{i=2}^{N} D_{2,i} = T_{2,N} - T_{2,1}$, $D_{(3)} \triangleq \sum_{i=2}^{N} D_{3,i} = T_{3,N} - T_{3,1}$, and $D_{(4)} \triangleq \sum_{i=2}^{N} D_{4,i} = T_{4,N} - T_{4,1}$, respectively. Then (5.21) and (5.22) can be rewritten respectively as

$$D_{(2)} = D_{(1)} + P + \theta_s \left(D_{(1)} + P\right),$$
$$D_{(4)} = D_{(3)} + R - \theta_s \left(D_{(4)} - R\right),$$

where $P \triangleq X_N - X_1$ and $R \triangleq Y_N - Y_1$. Notice that using the set of the actual time stamps ($T_{1,i}^{(R)}$ and $T_{4,i}^{(R)}$) yields exactly the same results.

1. Exponential Delay Model

For exponential delays, X_N, X_1, Y_N, and Y_1 are assumed to be i.i.d. exponentially distributed RVs with mean α. Then P and R become zero mean *Laplace* distributed RVs with variance $2\alpha^2$, respectively. Thus, the joint PDF of P and R is given by

$$f_{P,R}(p,r) = \left(\frac{1}{2\alpha}\right)^2 e^{-\frac{1}{\alpha}(|p|+|r|)}.$$

The likelihood function becomes

$$L(\theta_{\mathrm{s}},\alpha) = \left(\frac{1}{2\alpha}\right)^2 e^{-\frac{1}{\alpha}\left(\left|\frac{D_{(2)}-D_{(1)}(1+\theta_{\mathrm{s}})}{1+\theta_{\mathrm{s}}}\right|+\left|\frac{D_{(4)}(1+\theta_{\mathrm{s}})-D_{(3)}}{1+\theta_{\mathrm{s}}}\right|\right)}.$$

Substituting $1/\theta'_{\mathrm{s}} - 1$ into θ_{s}, the likelihood function can be rewritten as

$$L(\theta'_{\mathrm{s}},\alpha) = \left(\frac{1}{2\alpha}\right)^2 e^{-\frac{1}{\alpha}\left(D_{(2)}|\theta'_{\mathrm{s}}-\beta|+D_{(3)}|\theta'_{\mathrm{s}}-\gamma|\right)},$$

where $\beta \triangleq D_{(1)}/D_{(2)}$ and $\gamma \triangleq D_{(4)}/D_{(3)}$. The estimate $\hat{\theta}'_{\mathrm{s}}$ maximizing the likelihood function is given by

$$\begin{aligned}
\hat{\theta}'_{\mathrm{s}} &= \arg\min_{\theta'_{\mathrm{s}}} \left(D_{(2)}|\theta'_{\mathrm{s}}-\beta| + D_{(3)}|\theta'_{\mathrm{s}}-\gamma|\right), \\
\hat{\theta}'_{\mathrm{s}} &= \arg\min_{\theta'_{\mathrm{s}}} \sum_{i=1}^{2} K_i \left|\theta'_{\mathrm{s}} - \delta_{(i)}\right|, \qquad (5.23)
\end{aligned}$$

where the order statistics $\left\{\delta_{(i)}\right\}_{i=1}^{2}$ are generated from the given observations $\{\beta,\gamma\}$, and K_i represents distance terms equal to either $D_{(2)}$ or $D_{(3)}$.

Let $\hat{j} = \arg\min_j \sum_{i=1}^{2} K_i \left|\delta_{(j)} - \delta_{(i)}\right|$, then from the result in Appendix C, the

proposed MLLE can be expressed as

$$\hat{\theta}_{\mathrm{s}}^{EMLLE} = \frac{2}{\delta_{(1)} + \delta_{(2)}} - 1. \tag{5.24}$$

In the sequel, using the set of distances, the proposed MLLE for exponential random delays (EMLLE) can be rewritten as

$$\hat{\theta}_{\mathrm{s}}^{EMLLE} = \frac{2}{\beta + \gamma} - 1 = \frac{2D_{(2)}D_{(3)}}{D_{(1)}D_{(3)} + D_{(2)}D_{(4)}} - 1. \tag{5.25}$$

Now we are interested in the lower bound of the EMLLE to evaluate its asymptotic behavior. The derivative of the log likelihood function becomes

$$\frac{\partial \ln L\left(\theta'_{\mathrm{s}}, \alpha\right)}{\partial \theta'_{\mathrm{s}}} = \frac{D_{(2)}}{\alpha}\mathrm{sgn}\left(\theta'_{\mathrm{s}} - \beta\right) + \frac{D_{(3)}}{\alpha}\mathrm{sgn}\left(\theta'_{\mathrm{s}} - \gamma\right). \tag{5.26}$$

Then the expected value of the square of (5.26) is given by

$$\begin{aligned} E\left[\left(\frac{\partial \ln L\left(\theta'_{\mathrm{s}}, \alpha\right)}{\partial \theta'_{\mathrm{s}}}\right)^2\right] &= E_{P,R}\left[\frac{D_{(2)}^2 + D_{(3)}^2 + 2D_{(2)}D_{(3)}\mathrm{sgn}\left(\theta'_{\mathrm{s}} - \beta\right)\mathrm{sgn}\left(\theta'_{\mathrm{s}} - \gamma\right)}{\alpha^2}\right] \\ &\overset{(c)}{=} \frac{D_{(1)}^2 + D_{(4)}^2 + 4\alpha^2}{\alpha^2}, \end{aligned}$$

where (c) is due to the fact that P and R are independent. Therefore, the lower bound of the EMLLE is given by

$$\mathrm{var}(\hat{\theta}_{\mathrm{s}}^{EMLLE}) \geq \frac{\left(\frac{\partial \theta_{\mathrm{s}}}{\partial \theta'_{\mathrm{s}}}\right)^2}{E\left[\left(\frac{\partial \ln L(\theta'_{\mathrm{s}}, \alpha)}{\partial \theta'_{\mathrm{s}}}\right)^2\right]} = \frac{\alpha^2\left(1 + \theta_{\mathrm{s}}\right)^2}{D_{(1)}^2 + D_{(4)}^2 + 4\alpha^2}. \tag{5.27}$$

In fact, we have followed the same steps used in CRB derivation since the same reasoning and proof can be also applied to the lower bound derivation for the MLLE.

2. Gaussian Delay Model

Similarly, assuming X_N, X_1, Y_N, and Y_1 are i.i.d. normal distributed RVs with variance σ^2, P and R become zero mean normal distributed RVs with variance $2\sigma^2$, respectively. Then the joint PDF of P and R is given by

$$f_{P,R}(p,r) = \left(\frac{1}{4\pi\sigma^2}\right)^2 e^{-\frac{1}{4\sigma^2}\left(p^2+r^2\right)}.$$

Hence, the likelihood function becomes

$$L\left(\theta'_{\mathrm{s}},\sigma^2\right) = \left(\frac{1}{4\pi\sigma^2}\right)^2 e^{-\frac{1}{4\sigma^2}\left[D_{(2)}^2(\theta'_{\mathrm{s}}-\beta)^2+D_{(3)}^2(\theta'_{\mathrm{s}}-\gamma)^2\right]}.$$

Differentiating the log-likelihood function with respect to θ'_{s} yields

$$\frac{\partial^2 \ln L\left(\theta'_{\mathrm{s}},\sigma^2\right)}{\partial\theta'^2_{\mathrm{s}}} = -\frac{1}{2\sigma^2}\left[D_{(2)}^2\left(\theta'_{\mathrm{s}}-\beta\right)+D_{(3)}^2\left(\theta'_{\mathrm{s}}-\gamma\right)\right].$$

Thus the proposed MLLE for the Gaussian delay model (GMLLE) is given by

$$\hat{\theta}_{\mathrm{s}}^{GMLLE} = \frac{1}{\hat{\theta}'_{\mathrm{s}}} - 1 = \frac{D_{(2)}^2+D_{(3)}^2}{D_{(1)}D_{(2)}+D_{(3)}D_{(4)}} - 1. \tag{5.28}$$

Again, similar procedures can be applied to derive a lower bound for the GMLLE. The 2nd order derivative of the log likelihood function becomes

$$\frac{\partial^2 \ln L\left(\theta'_{\mathrm{s}},\sigma^2\right)}{\partial\theta'^2_{\mathrm{s}}} = -\frac{D_{(2)}^2+D_{(3)}^2}{2\sigma^2}. \tag{5.29}$$

The expected value of (5.29) is given by

$$E\left[\frac{\partial^2 \ln L\left(\theta'_{\mathrm{s}},\sigma^2\right)}{\partial\theta'^2_{\mathrm{s}}}\right] = -\frac{E\left[D_{(2)}^2+D_{(3)}^2\right]}{2\sigma^2} = -\frac{D_{(1)}^2+D_{(4)}^2+4\sigma^2}{2\sigma^2}.$$

Finally, the lower bound of the GMLLE is given by

$$\mathrm{var}(\hat{\theta}_{\mathrm{s}}^{GMLLE}) \geq \frac{\left(\frac{\partial\theta_{\mathrm{s}}}{\partial\theta'_{\mathrm{s}}}\right)^2}{-E\left[\frac{\partial^2 \ln L(\theta'_{\mathrm{s}},\sigma^2)}{\partial\theta'^2_{\mathrm{s}}}\right]} = \frac{2\sigma^2\left(1+\theta_{\mathrm{s}}\right)^2}{D_{(1)}^2+D_{(4)}^2+4\sigma^2}. \tag{5.30}$$

Note that the complexity of the MLLEs is far less than that of the GMLE. In fact, for the GMLE, the number of required multiplications and additions are about $4N+6$ and $10N$, respectively. While, both MLLEs require only a few multiplications and additions (less than 5) regardless of the number of beacons N. Moreover, for the GMLE, the fixed portion of delays d must be also estimated, which requires additional computations.

3. Combination of Clock Offset and Skew Estimation

Since the proposed MLLEs are only for estimating the clock skew θ_{s}, we still need to estimate the clock offset θ_{o} for a complete clock synchronization. Considering the given clock skew model, the ith observations of delays of timing message exchange U_i $(= T_{2,i} - T_{1,i})$ and V_i $(= T_{4,i} - T_{3,i})$ can be rewritten, respectively, as

$$U_i = d + X_i + \theta_{\mathrm{s}} \left(T_{1,i} + d + X_i\right) + \theta_{\mathrm{o}},$$
$$V_i = d + Y_i - \theta_{\mathrm{s}} \left(T_{4,i} - d - Y_i\right) - \theta_{\mathrm{o}}.$$

Since $T_{2,i}$ and $T_{4,i}$ are known values and θ_{s} can be estimated using the MLLE, the set of delay observations between two nodes can be recomposed by

$$U_i' = U_i - \hat{\theta}_{\mathrm{s}} T_{1,i} \quad \left(= d' + \theta_{\mathrm{o}} + X_i'\right), \tag{5.31}$$
$$V_i' = V_i + \hat{\theta}_{\mathrm{s}} T_{4,i} \quad \left(= d' - \theta_{\mathrm{o}} + Y_i'\right), \tag{5.32}$$

where $X_i' = (1+\theta_{\mathrm{s}}) X_i$, $Y_i' = (1+\theta_{\mathrm{s}}) Y_i$, and $d' = (1+\theta_{\mathrm{s}}) d$, respectively. Notice that it can be applied to the same clock offset estimator as in (5.1) and (5.7) for exponential and Gaussian delay models, respectively. Substituting the sets of delay

observations yields the following clock offset estimators:

$$\hat{\theta}_{\mathrm{o}} = \frac{\min\limits_{1\le i\le N} U_i' - \min\limits_{1\le i\le N} V_i'}{2} \qquad \text{(exponential delays)}, \tag{5.33}$$

$$\hat{\theta}_{\mathrm{o}} = \frac{\overline{U_i'} - \overline{V_i'}}{2} \qquad \text{(Gaussian delays)}. \tag{5.34}$$

Consequently, the proposed joint clock offset and skew estimators consist of the following steps:

1. Estimate clock skew using the proposed MLLE either $\hat{\theta}_{\mathrm{s}}^{EMLLE}$ or $\hat{\theta}_{\mathrm{s}}^{GMLLE}$ according to the type of random delays.

2. Recompose the sets of delay observations U_i' and V_i' as shown in (5.31) and (5.32).

3. Estimate clock time offset using either the estimator (5.33) or (5.34) corresponding to the given delay model.

In fact, the proposed MLLEs require multiple message exchanges in a sync period ($N > 1$) to obtain the set of distances ($\{D_{(i)}\}_{i=1}^{4}$). However, these estimators can be applied not only within the same sync period, but also throughout several consecutive sync periods. In other words, a new set of observations present in the next sync period can be substituted for the set of time stamps of the initial message exchange ($\{T_{i,N}\}_{i=1}^{4}$) in the initial sync period. This substitution can be sequentially performed thereafter. Therefore, the proposed MLLEs can be also applied to the single message exchange model ($N = 1$) like TPSN without further modifications. The performance of the MLLEs is analyzed in the following section.

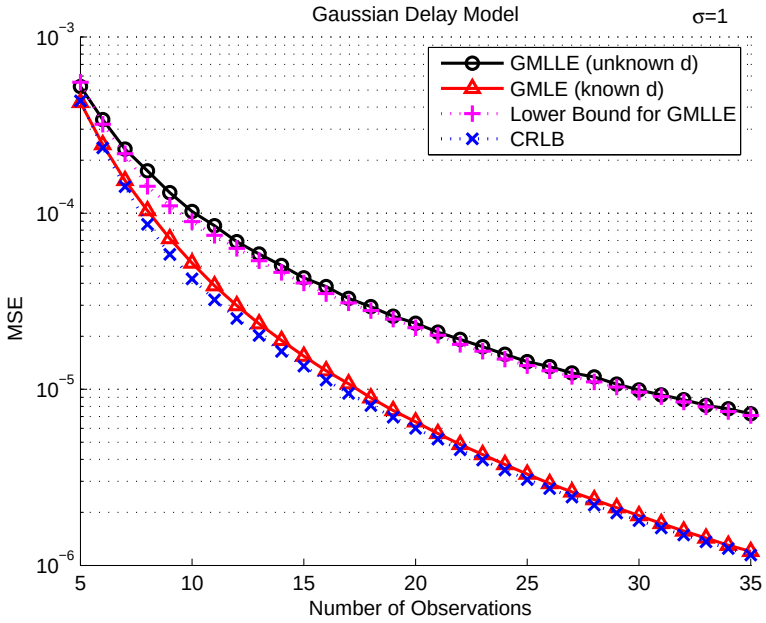

Fig. 14. MSE of the MLE of the Gaussian delay model (GMLE) and the Gaussian MLLE (GMLLE) for Gaussian random delays ($\sigma = 1$).

E. Simulation Results

Fig. 14 compares the mean-square error (MSE) of the GMLLE with the joint GMLE of clock skew and corresponding CRB when σ is 1. It can be seen that the GMLLE performs close to the GMLE for a reduced number of observations (N) (typical values for energy efficient regimes), and its variance goes to zero as the number of observations increases (consistent and asymptotically efficient). Note that the GMLLE works well without knowing the fixed portion of delays d, whereas the same is required by the joint GMLE.

Fig. 15 illustrates the MSE of the EMLLE relative to the joint GMLE in exponential random delay channels when α is 1. It can be seen that again the proposed MLLE is consistent and comparable to the GMLE. The consistency of the proposed MLLEs can be also checked from (5.27) and (5.30) since the corresponding MSE-

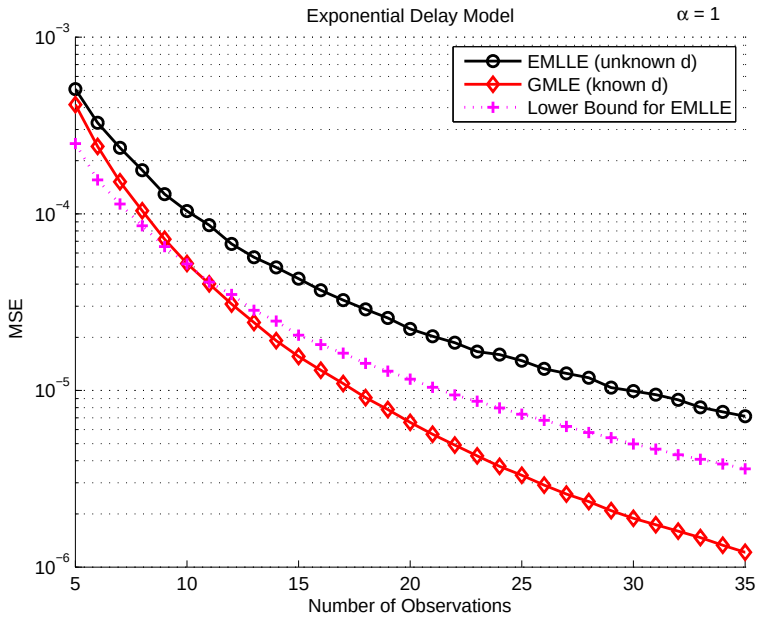

Fig. 15. MSE of the GLME and the exponential MLLE (EMLLE) for exponential random delays ($\alpha = 1$).

bounds approach 0 as N increases.

In order to evaluate the robustness of estimators, Fig. 16 compares the performance of the GMLE with the MLLEs in standard Gamma distributed (one of the most widely used models for capturing random queuing delays) random delay channels when γ is 2. Both MLLEs exhibit similar performance compared to the GMLE regardless of the type of random delays. This is due to the fact that the performance of the MLLE is dominated by the set of distances ($\{D_{(i)}\}_{i=1}^{4}$), which do not vary much with respect to the type of random delays.

Fig. 17 compares the performance of the proposed clock offset estimator (5.34) with the joint Gaussian MLE of clock offset derived in (5.15) in the Gaussian delay model when $\sigma = 0.5$. It can be seen that the joint MLE outperforms the proposed estimator due to the help of the prior knowledge of d and the complete set of

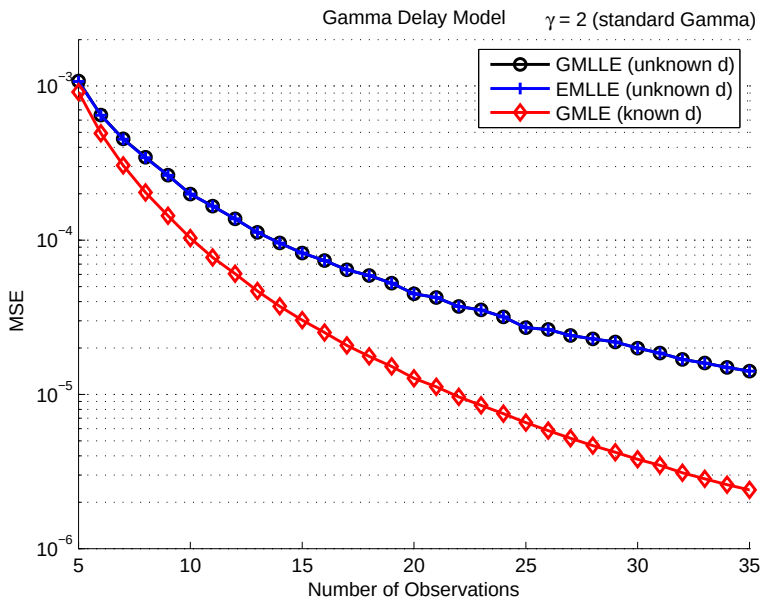

Fig. 16. MSE of the GLME and the MLLEs for Gamma random delays ($\gamma = 2$).

timestamps.

F. Conclusions

In this study, we have first derived the CRB for the well-known MLE of clock offset in TPSN assuming no clock skew, and normally and exponentially distributed delays, respectively. Then, using a more realistic clock model, the joint MLE of clock offset and skew has been proposed for Gaussian delays assuming the fixed portion of delays d is known. Furthermore, we proposed novel ML-like estimators, requiring no prior knowledge of d, for both Gaussian and exponential random delays, respectively. The proposed MLLEs can be implemented using simple modifications and present remarkably low complexity, which is an attractive feature for WSNs. These estimators and the derived performance bounds are targeting practical applications,

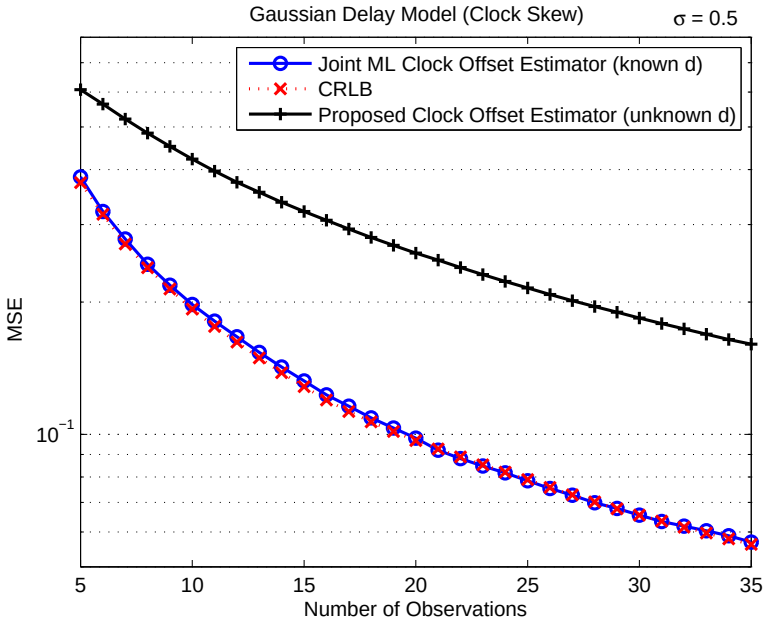

Fig. 17. MSE of the joint ML clock offset estimate and the proposed estimator for Gaussian random delays ($\sigma = 0.5$).

and significant steps are conducted towards assessing the performance of different protocols currently popular for synchronization in WSNs. The proposed joint GMLE and MLLEs can be applied without further modifications to any clock synchronization protocols based on two-way timing message exchanges. The contributions of this study are summarized in Table I. Future works include assessing the performance of other popular synchronization protocols to achieve a global performance analysis of existing protocols.

Table I. Contributions on clock synchronization protocols using two-way message exchanges for WSNs

<table>
<tr><th>Delay \ Clock</th><th>Clock Offset</th><th>Clock Offset and Skew</th></tr>
<tr><td>Exponential Delay</td><td>CRB in (5.5)</td><td>(unknown d) MLLE in (5.25) and (5.33)</td></tr>
<tr><td rowspan="2">Gaussian Delay</td><td rowspan="2">MLE in (5.7),
CRB in (5.8)</td><td>(unknown d) MLLE in (5.28) and (5.34)</td></tr>
<tr><td>(known d) joint MLE in (5.15) and (5.16),
CRB in (5.19) and (5.20)</td></tr>
</table>

CHAPTER VI

PAIRWISE BROADCAST SYNCHRONIZATION

A. Motivations

As discussed in Chapter II, there are a number of key factors in designing time synchronization protocols for WSNs, such as accuracy, energy consumption, scalability, acquisition time, implementation complexity, and robustness. The most important and crucial factor is the tradeoff between the accuracy and energy consumption (complexity). Increasing the synchronization accuracy requires in general more energy consumption for transmitting the RF timing messages among sensor nodes. On the other hand, the energy consumption for synchronization should be kept as small as possible since the power resources of common wireless sensors are strictly limited and not rechargeable in general. However, for most of the existing synchronization protocols, there is a lack of in-depth analysis to assess the energy-efficiency tradeoff of synchronization algorithms. We propose a new time synchronization schemes referred to as the Pairwise Broadcast Synchronization (PBS) protocol which efficiently combines both SRS and ROS approaches (described in Chapter III) to achieve network-wide synchronization with a significantly reduced number of synchronization messages, i.e., with lesser energy consumption.

The rest of the chapter is organized as follows. In Section B, we briefly summarize the way how to achieve network-wide synchronization for single-cluster sensor networks based on the newly developed approach, ROS, described in Chapter III. Section C analyzes the performance of PBS and compares it with those of other well known protocols. For the extension to general multi-cluster sensor networks, Section D proposes the network-wide pair selection algorithm and the group-wise pair

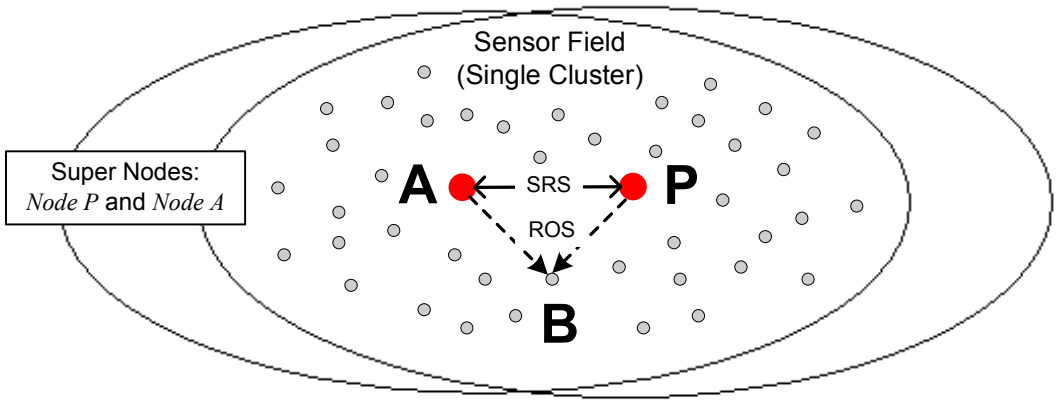

Fig. 18. PBS for single-cluster networks.

selection algorithm to select the best synchronization sequence aiming at minimizing the overall energy consumption, respectively. Section E analyzes the performance of the proposed pair selection algorithms with respect to the number of required synchronization messages (i.e., energy consumption). Finally, Section F summarizes and concludes this study.

B. Synchronization for Single-Cluster Networks

In Fig. 18, every node in a single-cluster network (e.g., *Node B*) can receive messages from both super nodes *Node P* and *Node A*, while *Node P* and *Node A* perform a pairwise synchronization using two-way timing message exchanges as shown in Chapter III. Therefore, the proposed PBS achieves global synchronization by performing a pairwise synchronization between the two super nodes using the ROS approach, and the joint clock offset and skew estimator for Gaussian random delays, derived in (3.12), could be used in regard. Moreover, it was shown that there is no difference between PBS and one of the most popular synchronization protocols, RBS, with regard to the accuracy of synchronization.

C. Comparisons and Analysis

This section compares the proposed PBS protocol with other well-known synchronization protocols, such as TPSN, RBS, and FTSP, with respect to the amount of energy consumption (number of required timing messages) and the synchronization accuracy. Let $N_{\mathbf{RBS}}$, $N_{\mathbf{TPSN}}$, $N_{\mathbf{FTSP}}$, and $N_{\mathbf{PBS}}$ denote the numbers of required timing messages for synchronization in PBS, RBS, TPSN, and FTSP, respectively.

In TPSN, since every node in the network is connected to its parent node except the reference node, there are $L-1$ branches (edges) in a hierarchical tree, where L is the overall number of sensor nodes [14]. Besides, for TPSN, $2N$ timing messages are required in every pairwise synchronization. Hence, $N_{\mathbf{TPSN}}$ is equal to the number of pairwise synchronizations times the number of required timing messages per pairwise synchronization, and therefore $N_{\mathbf{TPSN}} = 2N(L-1)$. This result can be applied to other level-based SRS protocols without loss of generality. The reference node must broadcast the beacon packet N times in RBS. Besides, every sensor node must send time readings upon receiving the broadcast beacons to all the other nodes in the network to compensate the relative clock offsets among each other [27]. Thus, $N_{\mathbf{RBS}} = N+L(L-1)/2$, since the number of unique pairs in the network is $L(L-1)/2$. In FTSP, each sensor node must send its timing messages once upon receiving timing messages from another sensor due to its flooding-based communication procedure [22]. Hence, the number of required timing messages in FTSP becomes: $N_{\mathbf{FTSP}} = NL$.

It is remarkable that the required number of timing messages for all the above mentioned protocols is proportional to the number of sensors in the network L or its square L^2. However, as discussed in Chapter III, PBS requires only $2N$ timing messages in every synchronization period since it adopts the energy efficient synchronization approach (ROS), i.e., $N_{\mathbf{PBS}} = 2N$. Hence, $N_{\mathbf{PBS}}$ does not depend on the

number of sensors in the network, which incurs an enormous amount of energy saving. Moreover, this gain increases proportionally with respect to the scale of the network. Consequently, the benefit of PBS over RBS, TPSN, and FTSP is clear and huge in terms of energy consumption with the cost of allocating 2 super nodes in the network. Note that RBS also requires a super node which broadcasts the reference beacons to all the other nodes in the network.

In case that there are other sensor nodes which are located outside of the checked region in Fig. 1, likewise RBS, the network could be divided into a number of separated groups (clusters) and they could be synchronized by additional pairwise synchronizations among super nodes in different groups, i.e., global synchronization can be achieved by a sequence of pairwise synchronizations. Here, diverse grouping and pair selection algorithms can be considered according to the type of the network. For instance, assuming that the level hierarchy of the network is discovered by an appropriate searching algorithm (e.g., as in [14]), there exist groups of parents and children nodes, where a group consists of a parent and its children nodes. Here, every parent node can investigate the connectivity among its children nodes and select the best sequence of synchronization pairs in order to minimize the required number of pairwise synchronizations, which maximizes the number of nodes performing ROS. Note that no network-wide heuristic connectivity search is required in this case because of its limited and known set of scanning nodes. The detailed extension of these preliminary considerations for the proposed PBS scheme is presented in the following sections.

The synchronization accuracy is another crucial designing factor to be concerned with. In general, it depends on a variety of different factors, such as the network platform and setup, channel status, and estimation schemes. The performance of existing protocols has been compared in terms of the synchronization accuracy in various references, e.g., [1], [13], [22], and [51]. As shown in Chapter III, the accuracy

of PBS is exactly the same as that of RBS. The interested reader is referred to the above mentioned references for additional insights regarding this issue.

When there are multiple synchronization clusters (groups) in the network, the proposed PBS requires a series of pairwise synchronizations to achieve network-wide synchronization, i.e., an independent pairwise synchronization is necessary in every level of the network. Hence, the performance of ROS at each level (depth) of the network is independent from that of ROS at another level. It has been experimentally shown that the time synchronization error in RBS follows as Gaussian distribution [27]. Here, we assume the clock estimation error of ROS is also Gaussian distributed due to its similarity as RBS. Then, the cumulative network-wide synchronization error can be modeled as a sum of normal RVs, i.e., thus another normal RV. From (3.14), the variance of the network-wide synchronization error can be approximated as

$$\mathrm{var}(\hat{\theta}_{\mathrm{o}}) \approx \sum_{i=1}^{d_{\max}} \mathrm{var}(\hat{\theta}_{\mathrm{o}}^{(i)}) \geq \sum_{i=1}^{d_{\max}} \mathrm{CRLB}(\hat{\theta}_{\mathrm{o}}^{(i)}), \tag{6.1}$$

where $\mathrm{CRLB}(\hat{\theta}_{\mathrm{o}}^{(i)})$ denotes the CRLB for the clock offset estimator at the ith level of the network and $d_{\max}$ denotes the maximum depth (level) of the network.

To achieve a network-wide global synchronization, the following crucial question should be solved: How to select the optimum set of pairwise synchronizations to minimize the number of timing message exchanges? The next sections answer this question and show a way how to guarantee network-wide synchronization.

D. Synchronization for Multi-Cluster Networks

There are two possible scenarios to extend the proposed PBS to the general multi-cluster network. When there is no problem with the placement of super nodes in the right positions of the network, the whole sensor field can be divided into several

clusters, where each cluster contains two individual super nodes whose communication ranges cover the entire cluster. Hence, every cluster can be first synchronized by performing a pairwise synchronization between a pair of super nodes. Then, likewise RBS, the global synchronization can be achieved by additional message exchanges (based on SRS) among super nodes in different clusters. In this case, the extension of PBS becomes mostly the problem of network implementation just like cell-planing problems in mobile communication networks.

However, if either deploying super nodes or deploying them where we want them placed is not available, there is no way to apply the above mentioned procedures. For this general scenario, this chapter proposes an energy-efficient pair selection algorithm, named the group-wise pair selection algorithm (GPA), to achieve global synchronization using ROS. Next, we first show a way to achieve global synchronization based on the network-wide heuristic search in order to reveal some general ideas of the pair selection problem as a preliminary study. Then, the proposed GPA is presented in detail.

1. Network-Wide Pair Selection Algorithm

Considering the energy-efficiency in time synchronization, the problem of finding the optimum set of pairwise synchronizations is equivalent to that of minimizing the number of overall pairwise synchronizations in the network. There are two fundamental criteria to select the best synchronization pairs. First, a pair of nodes containing the maximum number of nodes in their common coverage region of the pairwise synchronization has to be chosen during each selection step of the synchronization pair. Second, the depth (level) of the pairwise synchronization, i.e., the number of required successive pairwise synchronizations that is necessary to reach the reference node, has to be minimized since in general the synchronization errors increase with the

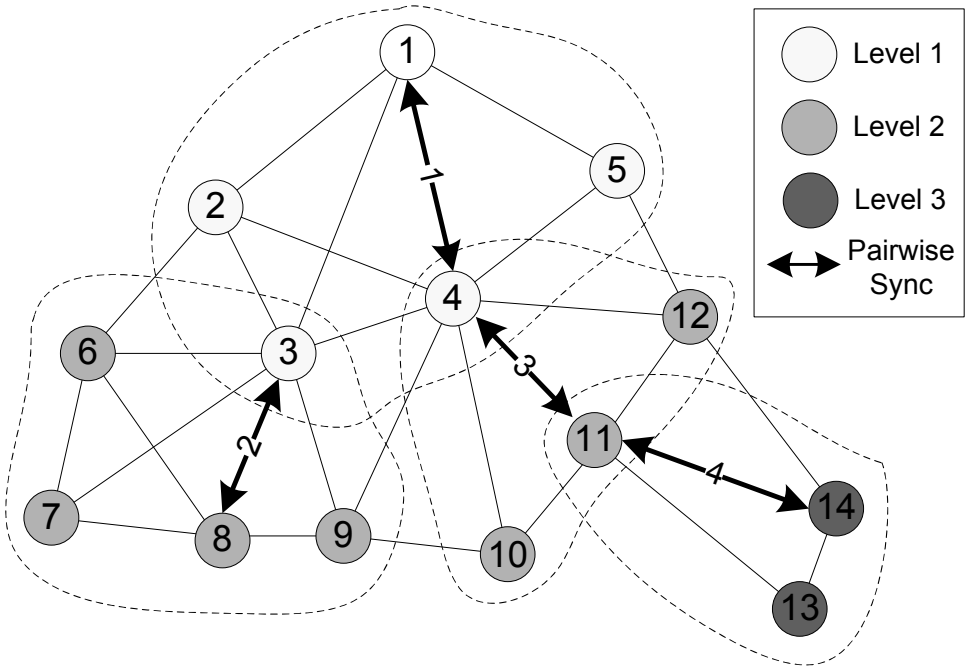

Fig. 19. Network connection hierarchy.

depth of the network as shown in (6.1). To find the best pair, the information about the network hierarchy and connectivity, which can be obtained by beacon exchanges among nodes, is required. Notice further that the network connection hierarchy can be constructed by applying the well-known breath-first search algorithm [55]. Here, every node in the network is required to send messages with their maximum power level satisfying a certain energy constraint.

For a graphical illustration of the proposed algorithms, Fig. 19 shows an example of a network connection hierarchy. The pairwise synchronization begins with the reference node *Node* 1, and four different branches (edges) are connected to the reference, i.e., there are four different nodes which can be chosen as the first synchronization pair. As mentioned, the criterion of selecting the best pair is to find a pair of nodes maximizing the number of synchronizing nodes (based on the ROS approach) from the pairwise synchronization. Let $p_{i,j}$ denote the pairwise synchronization between

Node i and *Node j*, $\mathbf{p}$ represent the pairwise synchronization sequence vector whose elements are a set of $p_{i,j}$. Define also by $N_{\mathbf{ROS}}^{i,j}$ the number of synchronizing nodes, which are performing ROS from $p_{i,j}$. In Fig. 19, *Node 4* must be selected as the first pair node since $N_{\mathbf{ROS}}^{1,4} = 3$ and it represents the maximum achievable value among all possible choices (all the other nodes in level 1, *Nodes* 2, 3, and 5, can be synchronized from $p_{1,4}$). The same criterion can be applied to determine the next pair of nodes thereafter, until all the nodes in the network are synchronized. Therefore, $p_{3,8}$, $p_{4,11}$, and $p_{11,14}$ are chosen as the second, third and fourth pairs, respectively. Consequently, a sequence of pairwise synchronizations is chosen to maximize the number of nodes performing ROS. In this example, the pairwise synchronization sequence vector is given by $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}, p_{11,14}\}$.

Considering the given pair selection criteria, we present next the Network-wide Pair Selection Algorithm (NPA) to find a pairwise synchronization sequence as a preliminary example. The network can be represented as a graph $G = (V, E)$, where V represents the set of nodes (vertices) and E stands for the set of edges (branches), whose elements are 2-element subsets of V, e.g., in Fig. 19, $V = \{s_i\}_{i=1}^{14}$. Assuming L_i denotes the subset of nodes located on level (depth) i, then $L_0 = \{s_1\}$, $L_1 = \{s_i\}_{i=2}^{5}$, $L_2 = \{s_i\}_{i=6}^{12}$, and $L_3 = \{s_{13}, s_{14}\}$ for the example depicted by Fig. 19. Let S denote a set of synchronized nodes whose initial element is $S = \{s_1\}$, and $M_{i,j}$ denote the ith row and jth column element of the adjacency matrix M of the graph G, where $M_{i,j} = 1$ when *Node i* and *Node j* are connected, and $M_{i,j} = 0$ otherwise.

Note that an arbitrary node *Node k* can be synchronized from $p_{i,j}$ if and only if *Nodes i* and *j* are connected and *Node k* is connected to both *Nodes i* and *j*, i.e., $M_{i,j} = M_{i,k} = M_{j,k} = 1$. Besides, the level of the nodes in a synchronization pair must differ by one level. Therefore, the number of synchronizing nodes from $p_{1,i}$ ($N_{\mathbf{ROS}}^{1,i}$) is

given by

$$N_{\mathbf{ROS}}^{1,i} = \sum_{j \neq i} M_{1,i} \cdot M_{1,j} \cdot M_{i,j} \qquad \forall\ s_i \notin S,\ s_j \notin S,$$

where $s_i \in L_1$ and $s_j \in L_1$. Hence, the first pair node can be obtained by maximizing $N_{\mathbf{ROS}}^{1,i}$:

$$\hat{i} = \arg\max_i N_{\mathbf{ROS}}^{1,i},$$

where $s_i \in L_1$, otherwise no connection exists between *Node* 1 and *Node* i. In the example of Fig. 19, $\hat{i} = 4$ because $N_{\mathbf{ROS}}^{1,i}$ is 3 and achieves the maximum value. Thus, $p_{1,4}$ is selected as the first pair. Note that a pair of nodes in the same level should not be selected as a valid pair in order to limit the bound for the maximum synchronization error which is proportional to the depth of the pairwise synchronization tree according to the second selection criterion. Hence, in general, to find the second pair of nodes in this example, another node in L_1 should be chosen until all the nodes in L_1 are synchronized. However, in this example, there are no remaining unsynchronized nodes in L_1 after $p_{1,4}$ since all the nodes in L_1 are already synchronized by $p_{1,4}$ ($S = \{L_0, L_1\}$).

The same maximization procedure can be applied to find the next synchronization pair. Similarly, a general formula for finding $N_{\mathbf{ROS}}^{i,j}$ is given by

$$N_{\mathbf{ROS}}^{i,j} = \sum_{k \neq j} M_{i,j} \cdot M_{i,k} \cdot M_{j,k} \qquad \forall\ s_i \in S,\ s_j \notin S,\ s_k \notin S, \tag{6.2}$$

where s_i is a candidate of the next parent node and the levels of s_j and s_k are equal and a level higher than that of the parent node. Again, based on the selection criterion, a node in a lower level has priority to be chosen as a parent node of the next pairwise synchronization. Likewise, the next synchronization pair can be found by maximizing $N_{\mathbf{ROS}}^{i,j}$:

$$(\hat{i}, \hat{j}) = \arg\max_{i,j} N_{\mathbf{ROS}}^{i,j}. \tag{6.3}$$

Here, $p_{\hat{i},\hat{j}}$ becomes the next element of $\mathbf{p}$ and all synchronized nodes from $p_{\hat{i},\hat{j}}$ are added to S. From (6.2) and (6.3), the second synchronization pair becomes $p_{3,8}$ in this example since $N_{\mathbf{ROS}}^{3,8}$ is 4 and maximum among all possible combinations of i and j. Thus, $\mathbf{p}$ becomes $\{p_{1,4}, p_{3,8}\}$ and $S = \{L_0, L_1, \{s_i\}_{i=6}^{9}\}$. Likewise, the third pair is chosen to be $p_{4,11}$, $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}\}$, and $S = \{L_0, L_1, L_2\}$. Repeating the same procedure (here, $s_i \in L_2$) gives $p_{11,14}$ as the last synchronization pair, and hence a complete sequence becomes $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}, p_{11,14}\}$ as depicted in Fig. 19. Fig. 20 illustrates NPA.

2. Group-Wise Pair Selection Algorithm

To discover the overall network connectivity, every single node in the network has to transmit the connection discovery beacons and send back acknowledgement packets upon receiving other beacons from its adjacent nodes (e.g., the breath-first search algorithm in [55]). For WSNs consisting of a large number of nodes, discovering the network connectivity is not a simple task and requires a number of packet exchanges in general. Therefore, we instead propose an efficient alternative method, the Group-wise Pair Selection Algorithm (GPA), which relies on the hierarchical structure (spanning tree) of the network to simplify the connection discovery procedure in NPA. Note that the hierarchical tree of the network can be generated by a level discovery procedure as discussed in [14]. Once a hierarchical tree is established, there exist groups of parents and children nodes, where a group consists of a parent and its children nodes. In GPA, instead of discovering the entire network connectivity, every parent node only investigates the connectivity among its children nodes. Therefore, the reference node does not need to find the pairwise synchronization sequence of the entire network, but need only to find the pairwise synchronization sequence among its children (level 1 nodes), and the other parent nodes successively perform

NETWORK-WIDE PAIR SELECTION ALGORITHM

Input: Graph (G), Adjacency matrix (M),

Maximum level/depth ($d_{\max}$)

Output: PS sequence vector ($\mathbf{p}$)

Initial values: $n = m = 1$

while $n \leq d_{\max} - 1$

  for all i, j, and k

  ($s_i \in S$, $s_i \in L_{n-1}$, and $s_j \notin S, s_k \notin S$, $s_j \in L_n, s_k \in L_n$)

    $N_{\mathbf{ROS}}^{i,j} \leftarrow \sum_{k \neq j} M_{i,j} \cdot M_{i,k} \cdot M_{j,k}$

  $(\hat{i}, \hat{j}) \leftarrow \arg\max_{i,j} N_{\mathbf{ROS}}^{i,j}$.

  $\mathbf{p}(m) \leftarrow p_{\hat{i},\hat{j}}$

  $m \leftarrow m + 1$

  If any j, $s_j \in L_n$ and $s_j \notin S$, exists

    then repeat from 2 to 6

    else $n \leftarrow n + 1$

$*$ $\mathbf{p}(m)$: mth element of $\mathbf{p}$

Fig. 20. Network-wide pair selection algorithm.

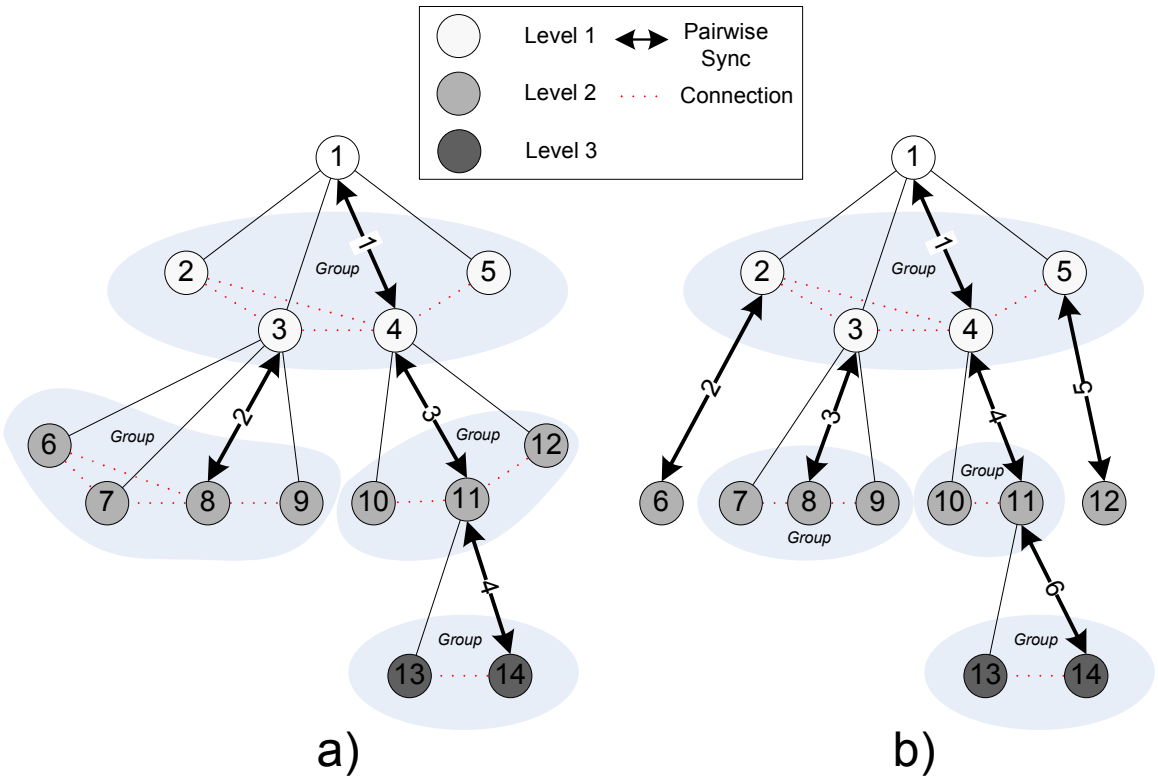

Fig. 21. Examples of hierarchical spanning trees of the network.

the same connection searching procedure as the reference node. As a result, GPA significantly reduces the complexity of building up a connection hierarchy, and requires a far smaller number of connection discovery beacons than NPA due to its limited and known set of scanning nodes. Furthermore, the work loads to find the best pairwise synchronization sequence can be balanced by sharing the roles of connection discovery and pair selection with the reference node and other parent nodes, i.e., no network-wide heuristic connection search is required for GPA.

Fig. 21 shows some possible hierarchical trees of the sample network. In Fig 21-a, the network can be synchronized using GPA with the same number of pairwise synchronizations as NPA, and *Nodes* 4, 8, 11, and 14 are chosen as parent nodes. However, the number of pairwise synchronizations for GPA depends on the specific

GROUP-WISE PAIR SELECTION ALGORITHM

Input: Graph (G) and adjacency matrix (M) of each group

Output: PS sequence vector ($\mathbf{p}$) of each group

Initial value: $n = 1$

1 **for each** group whose parent is s_i

2 **for all** j and k ($s_j \notin S, s_k \notin S$, and children of s_i)

3 $N_{\mathbf{ROS}}^{i,j} \leftarrow \sum_{k \neq j} M_{j,k}$

4 $\hat{j} \leftarrow \arg\max_j N_{\mathbf{ROS}}^{i,j}$.

5 $\mathbf{p}(n) \leftarrow p_{i,\hat{j}}$

6 $n \leftarrow n + 1$

7 **If** any j, $s_j \notin S$, exists

8 **then** repeat from 2 to 6

$*$ $\mathbf{p}(n)$: nth element of $\mathbf{p}$

Fig. 22. Group-wise pair selection algorithm.

hierarchical tree, which is randomly constructed, and is greater than NPA in general. For instance, for another possible tree of the network as in Fig 21-b, the required number of pairwise synchronizations is 6 instead of 4. Note that, for the same example, TPSN requires 13 pairwise synchronizations, same as the overall number of branches (edges). The proposed GPA is presented in Fig 22, and the connection discovery process for GPA is summarized in the next subsection.

a. Group-Wise Connection Discovery

As the level discovery phase in TPSN [14], GPA first creates a hierarchical structure (spanning tree) of the network, then it searches the connection status among a set of children nodes in every parent-children group. The connection discovery procedure in GPA consists of the following steps:

1. Select a reference node using an appropriate leader election algorithm (or picks up a node having the highest priority) and assign it to a zero level.

2. Broadcast a level discovery packet containing the identity and the level of packet.

3. Every node who receives a level discovery packet assign its level to the next greater level (depth) than that of the received packet and sends a new level discovery packet attaching its own level.

4. Repeat this process until every node in the network successfully assigns a level.

5. Once a hierarchical tree is established, every parent-children group performs the following operation: every child node broadcasts a connection discovery packet to other children nodes and sends back acknowledgement packets upon receiving other connection discovery packets.

After being assigned a level, every node discards further packets requesting level discovery to prevent collisions. Besides, connection discovery packets from any children nodes belonging to other groups will be also discarded. Notice also that other algorithms can be considered when constructing the spanning tree [18], [56].

Fig. 23 compares the complexity of NPA to establish the network connection hierarchy with that of GPA, which assumes a level hierarchy, with respect to the

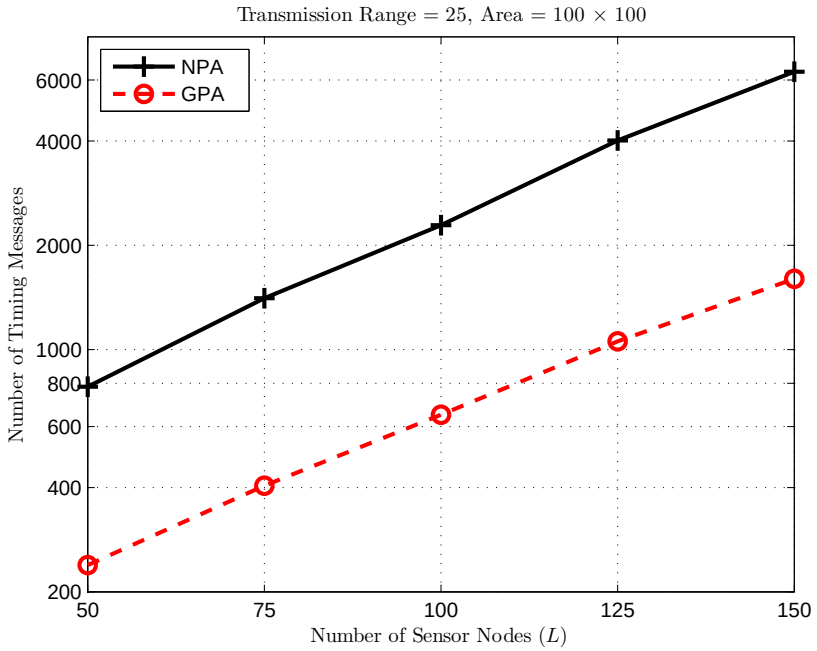

Fig. 23. Number of messages for constructing the network hierarchy (GPA vs NPA).

number of sensor nodes. In this simulation, sensors are randomly deployed in the area 100×100, the transmission range of each sensor is set to be 25, and the reference node is assumed to be located at the center of the simulation area. It can be seen that the complexity becomes greater as the number of sensor nodes (density) increases. The number of required discovery messages of NPA is about four times larger than that of GPA. Note that this Chapter assumes a static network, i.e., the network hierarchy does not need to be reconstructed frequently, i.e., a larger number of periodic re-synchronizations will be required based on the same network hierarchy. The following section analyzes the proposed algorithms in terms of the number of required timing messages, and compares them with other existing protocols.

E. Comparisons and Analysis

This section compares the proposed algorithms with other conventional ones such as TPSN, RBS, and FTSP in terms of the number of required timing messages, and thus predicts energy consumption, for network-wide synchronization. Assuming $|\mathbf{p}|$ denotes the number of elements in a pairwise synchronization sequence vector $\mathbf{p}$, then the total number of timing messages for NPA ($N_{\mathbf{NPA}}$) is given by

$$N_{\mathbf{NPA}} = 2N|\mathbf{p}|, \tag{6.4}$$

where N is the number of beacons per each node in a pairwise synchronization. Similarly, for GPA, the total number of timing messages ($N_{\mathbf{GPA}}$) is given by

$$N_{\mathbf{GPA}} = 2N \sum_{i=1}^{N_{\mathrm{G}}} |\mathbf{p}_i|, \tag{6.5}$$

where N_{G} denotes the number of parent-children groups and $\mathbf{p}_i$ denotes the pairwise synchronization sequence vector of the ith group. In the given example, $|\mathbf{p}| = 4$ (see Fig. 19) and $\sum_{i=1}^{N_{\mathrm{G}}} |\mathbf{p}_i| = 4$ or 6 (see Fig. 21-a and b), i.e., $N_{\mathbf{NPA}} = 8N$ and $N_{\mathbf{GPA}} = 8N$ or $12N$.

Lemma 1. Let $N_{\mathbf{TPSN}}$ be the required number of timing messages in TPSN, then $N_{\mathbf{TPSN}} = 2N(L-1)$, where L is the number of overall sensor nodes in the network.

Proof. Since every node in the network is connected to its parent node except a reference node, there are $L-1$ branches (edges) in a hierarchical tree. Besides, for TPSN, $2N$ timing messages are required in every pairwise synchronization. The number of required timing messages in TPSN is equal to the number of pairwise synchronizations times the number of required timing messages per pairwise synchronization, and therefore $N_{\mathbf{TPSN}} = 2N(L-1)$. □

Lemma 1 shows that $N_{\mathbf{TPSN}}$ is proportional to the number of overall nodes in the network. In the given example, $N_{\mathbf{TPSN}} = 26N$ which is greater than either $N_{\mathbf{NPA}}$ or $N_{\mathbf{GPA}}$. Note that this result can be applied to other level-based SRS protocols without loss of generality.

Lemma 2. Let $N_{\mathbf{FTSP}}$ be the number of required timing messages in FTSP, then $N_{\mathbf{FTSP}} = NL$.

Proof. For FTSP, every sensor node must send its time readings upon receiving beacons (or broadcast beacons) to other nodes so that they can estimate the relative clock offsets among each other. Therefore, the number of required timing messages in FTSP is equal to the number of sensor nodes times the number of beacons: $N_{\mathbf{FTSP}} = NL$. □

Lemma 3. Let $N_{\mathbf{RBS}}$ be the number of required timing messages in RBS, then $N_{\mathbf{RBS}} = N + L(L-1)/2$.

Proof. The reference node must broadcast the beacon packet N times in RBS. Besides, every sensor node must send time readings upon receiving the broadcast beacons with all the other nodes in the network to compensate relative clock offsets among each other [27]. Thus, $N_{\mathbf{RBS}} = N + L(L-1)/2$, since the number of unique pairs in the network is $L(L-1)/2$. □

For multi-cluster sensor networks consisting of super nodes (the former scenario), direct comparison with the proposed PBS and RBS is not available since they assume different network setups. For RBS, the network should be divided into a number of separated subgroups (clusters) such that every node in a subgroup is located within the transmission ranges of any other nodes in the same subgroup, i.e., a single hop topology is applied to each subgroup. However, PBS consists of a different set of

clusters, where each of them is the common coverage region of two super nodes. Indeed, both schemes requires extra timing message exchanges among subgroups for global synchronization. There exist nodes (could be super nodes) which are connected to multiple clusters and share global timing information to maintain network-wide synchronization. Thus, both $N_{\mathbf{PBS}}$ and $N_{\mathbf{RBS}}$ depend on the number of clusters and their connection status.

Assuming no super nodes in the network, in Fig. 24, the performances of $N_{\mathbf{NPA}}$ and $N_{\mathbf{GPA}}$ are compared with those of $N_{\mathbf{TPSN}}$ and the lower bound for $N_{\mathbf{RBS}}$ with respect to the number of overall sensor nodes. Again, in this simulation, sensor nodes are randomly deployed on an area of 100×100, the transmission range of each sensor is 25, and the reference node is assumed to be located at the center of the simulation area. The number of beacons (N) is set to be 10 in this simulation. It can be seen that PBS (with both GPA and NPA) requires a much lower number of timing messages than the other ones, such as TPSN, FTSP, and RBS, and the gaps between the required number of message transmissions of PBS and those of other protocols become greater as L increases. Therefore, for densely deployed WSN, PBS has a significant benefit in terms of energy consumption versus either TPSN or RBS. Besides, the proposed GPA performs quite close to NPA even though it does not require a heuristic network connection search. As mentioned, GPA can be implemented by simply adding a group-wise connection discovery procedure to the conventional level discovery process in an arbitrary level-based synchronization protocol like TPSN.

Fig. 25 evaluates the performance of the proposed algorithms with respect to the transmission range assuming the same simulation setup. The number of overall sensor nodes is fixed to 100 in this simulation. It can be seen that as the transmission range (density of the network) increases, $N_{\mathbf{GPA}}$ decreases (energy efficiency increases)

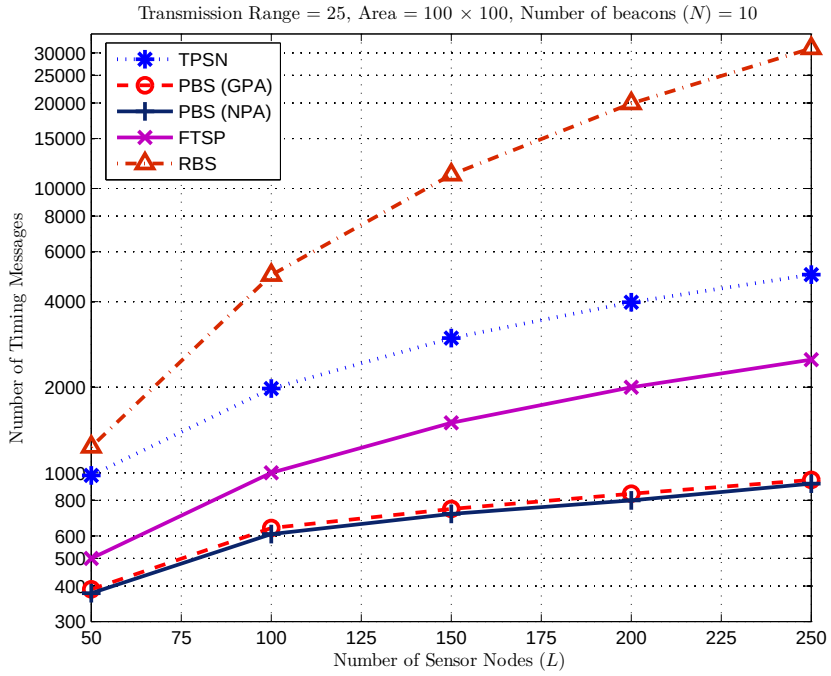

Fig. 24. Required number of message exchanges with respect to the number of sensor nodes.

since a larger number of sensor nodes are able to perform ROS.

F. Conclusions

In this study, a novel synchronization protocol has been proposed to reduce the overall energy consumption in synchronization based on a new synchronization approach that was called receiver-only synchronization. In the proposed Pairwise Broadcast Synchronization (PBS) protocol, a number of sensor nodes can be synchronized without any message transmission, i.e., they can be synchronized by only receiving timing messages between pairs of nodes. Thus, PBS significantly reduces the overall network-wide energy consumption by decreasing the number of required timing messages in synchronization. The simulation and analytical results showed that the proposed

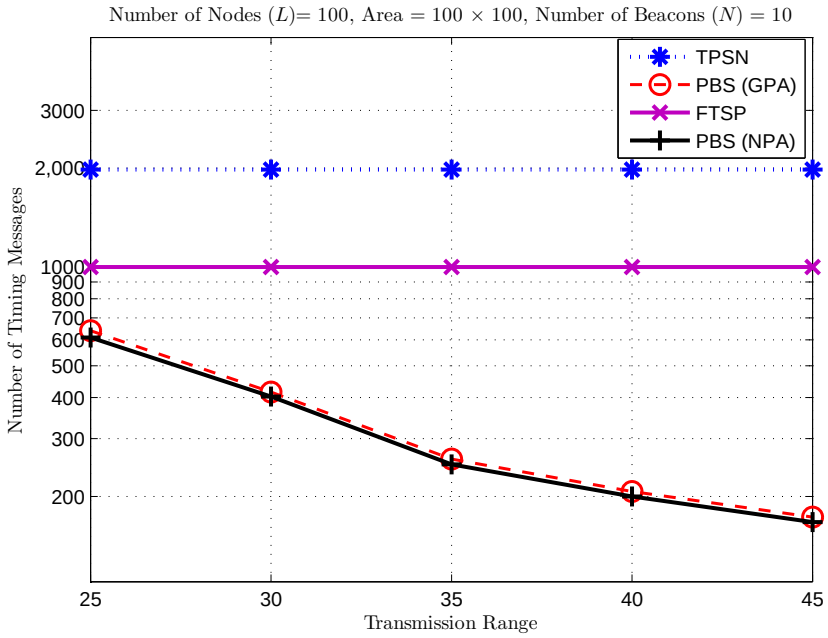

Fig. 25. Required number of message exchanges with respect to the transmission range.

scheme requires a far smaller number of timing messages than other well known protocols such as RBS, TPSN, and FTSP, and the benefits of the proposed scheme remarkably increase as the number of sensors increases or the sensors are densely deployed.

For the network consisting of multiple clusters, PBS first investigates a hierarchical connection tree of the network, then applies an energy-efficient pair selection algorithm, named group-wise pair selection algorithm (GPA), to achieve global synchronization. The proposed GPA only searches the connectivity among children nodes in every parent-children group of the spanning tree. Moreover, GPA can be easily combined with other level-based protocols by simply adding a group-wise connection discovery procedure.

This new approach and the main ideas presented herein could also be fully or partially applied to improve the performance of existing protocols or for designing

new protocols. Experimental performance evaluation and comparisons with other existing protocols represent an open research work for future.

CHAPTER VII

ADAPTIVE MULTI-HOP TIME SYNCHRONIZATION

A. Motivations

Developing long-term and network-wide timing-sync protocols that are energy-efficient represents one of the key strategies for the successful deployment of long-lived sensor networks. However, most of the existing protocols have focused only on achieving synchronization for short time-scales, and are not appropriate for long-term synchronization. Recently, the adaptive-clock synchronization protocols [52] and [53] considered optimizing the network synchronization protocol with the aim of achieving a specific synchronization accuracy with minimal energy consumption. The adaptive clock synchronization protocol [53] represents a probabilistic extension of RBS and proposes a mechanism for determining the minimum number of synchronization beacons and synchronization rate in order to achieve a pre-established clock synchronization error. Recently, Ganeriwal et al. [52] proposed for the first time a measurement-based study for designing an energy-efficient rate-adaptive long-time synchronization protocol (RATS) that adapts the synchronization period, number of beacons, and length of prediction window to achieve an application-specific accuracy.

Motivated in part by these preliminary contributions, we propose a more powerful adaptive multi-hop timing synchronization (AMTS) scheme with the goal of achieving a long-term network-wide synchronization with minimal energy consumption. AMTS exhibits a number of attractive features:

- It represents a significantly enhanced extension of TPSN aiming at minimizing the overall energy consumption in large-scale and long-lived sensor networks.
- It is equipped with flexible mechanisms to adjust the synchronization mode, the

period of network-wide timing synchronization (re-synchronization rate), and joint clock offset and skew estimators in order to achieve long term reliability of synchronization.

- It employs a sequential message exchange technique and an energy-efficient signaling scheme to further reduce the energy consumption in synchronization procedures.
- As opposed to RBS [27] and FTSP [22] that perform very poorly in high-latency acoustic networks, AMTS provides excellent performance in underwater acoustic networks characterized by high propagation delays and possible clock skew variations.

The rest of this chapter is organized as follows. In Section B, the main ideas and basic concepts of AMTS are provided. The following three sections from Section C to Section E present in a detailed manner all the three functional phases of AMTS such as level discovery, synchronization, and network evaluation phases, respectively. The performance of AMTS is simulated and compared with that of TPSN in Section F. Finally, Section G summarizes and concludes this study.

B. Main Ideas

AMTS is based on the similar system model as in TPSN and employees a number of novel features as well. It consists of three functional phases: *network level discovery phase*, *synchronization phase*, and *network evaluation phase*, and a number of network parameters such as latency factor, average number of hops, and re-synchronization period to optimize the synchronization protocol. Relative to TPSN, AMTS assumes the additional *network evaluation phase*, while the functions of the other two phases are similar to the ones encountered in TPSN.

Robustness to high-latencies and network delays is ensured based on the clock estimators presented in Chapter V, and therefore AMTS fits well for sensor network applications having large delays in timing message exchanges such as underwater acoustic sensor networks [54]. Besides, AMTS adapts the joint clock offset and skew estimators to increase the re-synchronization period. In addition, novel sequential message exchange and an efficient signaling technique are adopted in order to further decrease energy consumption.

As TPSN, generating a hierarchical structure in the network, the level discovery phase, is the first step of AMTS. In this phase, every single node in the network will be assigned a level and is ready for synchronization. The second step of AMTS, called the time synchronization phase, consists in pairwise synchronizations between adjacent nodes until every node in the network is synchronized to the reference. In the synchronization phase, AMTS estimates not only the current clock offset but also the clock frequency (skew) to guarantee long term reliability of synchronization while TPSN only estimates the clock offset. Hence, it requires far less frequent re-synchronization. Finally, the reference node investigates the current status of network traffic in order to optimize the re-synchronization period and the number of beacons in terms of energy efficiency. Besides it selects the synchronization mode between the *always on* (**AO**) mode (always maintain network-wide synchronization) and the *sensor initiated* mode (**SI**) (synchronize only when it needs to) based on the network status. This step stands for the network evaluation phase, and its goal is to minimize the number of message exchanges for synchronization in a given time, i.e., it aims to minimize total energy consumption for synchronization. AMTS periodically repeats the synchronization and network evaluation phases to minimize total energy consumption with respect to the current network status. The functional phases of AMTS are summarized below (with the second and third phases repeated periodically).

- *Level discovery phase*: It is the same as that in TPSN, and is used for generating a hierarchical structure in the network.
- *Synchronization phase*: It is similar to the corresponding synchronization phase in TPSN. However, as opposed to TPSN, AMTS adjusts not only the current clock offset but also the clock skew to guarantee the long term synchronization, while TPSN only estimates the clock offset. Hence, AMTS requires a far less frequent re-synchronization.
- *Network evaluation phase*: The reference node investigates the current status of network traffic in order to select the synchronization mode between the **AO** mode (always maintain network-wide synchronization) and the **SI** mode (synchronize only when it needs to). Besides, it optimizes the re-synchronization period and the number of beacons per each pairwise synchronization.

The following sections describe these phases in detail.

C. Level Discovery Phase

Similarly to TPSN, the role of level discovery phase in AMTS is to create a hierarchical structure (spanning tree) of the network. The level discovery phase in AMTS consists of the following steps: 1. Select a root node using an appropriate leader election algorithm and assign a zero level to the root node. 2. Broadcast a level discovery packet (LDP) containing the identity and the level of packet. 3. Every node who receives a LDP assign its level to one level greater than that of the received packet and sends a new level discovery packet attaching its own level. 4. Repeat this process until every node in the network successfully assigns a level. After being assigned a level, every node discards further packets requesting level discovery to prevent from collisions.

D. Synchronization Phase

This phase performs pairwise synchronization between a set of nodes by exchanging timing messages. For the **AO** mode, a series of pairwise synchronizations will take place until every node in the network is synchronized to the reference, i.e., the message exchanges are occurring at all branches of the network spanning tree. On the other hand, for the **SI** mode, only the nodes participating in the particular multi-hop data transmission synchronize with each other. AMTS adapts the joint clock offset and skew estimation mechanism proposed in Chapter V by considering the long-term reliability and energy-efficiency of synchronization as design criteria.

E. Network Evaluation Phase

In this phase, the network examines the total amount of message exchanges for synchronization during the last synchronization period, then adjusts the duration of the next synchronization period to minimize the overall energy consumption for synchronization. When the network traffic occurs rarely and synchronization delay is not a critical problem, applying the **SI** mode is a better choice to save network resources instead of using the **AO** mode. In addition, for some applications, the sensor clocks might be allowed to go out of synchronization unless sensing events happen. Another critical problem is to determine the required number of timing message exchanges (beacons) per each pairwise synchronization. To fulfill higher requirement of synchronization accuracy, a larger number of message transfers and corresponding signal processing is needed in each pairwise synchronization. However, as the number of required timing messages per each pairwise synchronization increases, the overall number of timing messages in a synchronization period increases. Hence, there is a tradeoff between accuracy and energy consumption.

1. Synchronization Mode Selection

To address these design challenges, AMTS determines several network parameters such as the synchronization mode (**AO** or **SI**), the re-synchronization period τ, and the number of beacons per pairwise synchronization N. Indeed, AMTS aims at efficient usage of network resources (i.e., energy saving) in synchronization. The idea of selecting the synchronization mode between **AO** and **SI** is based on the observation that when the network traffic occurs rarely and synchronization delay is not a critical problem, keeping all the sensor nodes synchronized all the times (**AO** mode) is not a good strategy since synchronization consumes a lot of energy. In addition, for some applications, the sensor clocks might be allowed to go out of synchronization unless sensing events happen. In this case, the **SI** mode, where only nodes participating in a particular multi-hop data transmission synchronize with each other, is a better choice. Here, we define the following parameters:

- B: Number of branches (edges) in a spanning tree of the network. It can be obtained after the level discovery phase.
- τ: (Re-) Synchronization period, i.e., the time between re-synchronization.
- $\overline{h}$: Average number of hops per unit time. In every sensing event, the destination node accumulates the number of hops that have occurred in that particular transmission to its storage. During the synchronization phase, the reference node collects the information about the total number of hops occurred in the last synchronization period and determines the average number of hops per unit time ($\overline{h}$) in the network. This information indicates how busy the network traffic is and can be included in timing messages with a small overhead.
- δ: Latency factor ($0 \leq \delta \leq 1$) reflecting the amount of allowed delay in data

transmission. A higher latency factor means less concern for network delays. For example, δ is set to be 0 for sensor networks requiring network synchronization all the time. On the other extreme, for delay-independent networks, δ should be close to 1.

- N: Number of timing message exchanges per pairwise synchronization.

As mentioned before, the goal of AMTS is to minimize the number of required timing messages. In the **AO** mode, the number of timing messages per unit time is given by $\overline{M} = 2BN/\tau$, while in the **SI** mode $\overline{M} = 2\overline{h}N$. To minimize the number of timing messages per unit time $\overline{M}$, the synchronization mode should be selected as follows:

$$\frac{2BN\delta}{\tau} \underset{\mathbf{SI}}{\overset{\mathbf{AO}}{\lessgtr}} 2\overline{h}N, \tag{7.1}$$

where the latency factor δ varies from 0 to 1 such that the more delay-dependent networks assume a larger value of δ and vise versa ($0 \leq \delta \leq 1$).

As the clock synchronization period τ increases, the network becomes more power efficient. Thus, τ should be chosen as large as possible. However, a too large value of τ induces a critical synchronization problem since the clock difference (offset) between nodes keeps generally increasing with time. Hence, there exists a maximum timing synchronization period (τ_{max}) which is determined by the oscillator regulations (hardware specifications) and the accuracy of estimators. Notice that sensing data transmission is not available during the synchronization phase (τ_{sync}), so the re-synchronization period is given by $\tau = \tau_{max} + \tau_{sync}$. In sequel, (7.1) can be rewritten as

$$\tau \underset{\mathbf{SI}}{\overset{\mathbf{AO}}{\gtrless}} \frac{B\delta}{\overline{h}}. \tag{7.2}$$

From (7.2), the synchronization mode changes from **AO** into **SI** when τ is smaller than $B\delta/\overline{h}$ and vise versa. In the **SI** mode, the reference node periodically asks

the number of hops that occurred during the past time interval, and then makes a decision whether or not to switch to the **AO** mode. Actually, τ is also dependent on N since it strongly depends on the accuracy of timing offset estimators. A more detailed analysis of τ is provided in the following subsection.

2. Determination of Synchronization Period (τ)

As the re-synchronization period τ increases, the network becomes more power efficient. Thus, τ should be chosen as large as possible. However, a too large value of τ induces a critical synchronization problem since the clock difference (offset) between nodes keeps generally increasing with time. Hence, there exists a maximum re-synchronization period ($\tau_{\max}$) which is determined by the oscillator specifications and the accuracy of estimators.

Suppose that the clock timing mismatch ε between the two nodes is modeled as follows: $\varepsilon = \varepsilon_o + \varepsilon_s t$, where t denotes the reference time, ε_o and ε_s stand for the clock offset and skew errors, respectively. Let $\varepsilon_{o,i}$ and $\varepsilon_{s,i}$ denote the clock offset and skew estimation errors when i message exchanges occur between the two nodes. In general, it is difficult to determine any specific mathematical model for either clock offset or skew errors. Herein, we model both clock offset and skew errors by normal distributions based on the experimental results reported in [14] and [27]:

$$\varepsilon_{o,i} \sim \mathcal{N}(0, \sigma^2_{\varepsilon_{o,i}}) \quad 1 \le i \le N,$$
$$\varepsilon_{s,i} \sim \mathcal{N}(0, \sigma^2_{\varepsilon_{s,i}}) \quad 1 \le i \le N,$$

where $\varepsilon_{o,i}$ and $\varepsilon_{s,i}$ denote the clock skew and offset estimation errors after the ith message exchanges, respectively. Note that clock skew estimation is only available when there are multiple message exchanges. Hence, $\varepsilon_{s,1}$ stands for the clock skew error when no skew estimation occurred. Here, the maximum clock mismatch can be

modeled as another normal distribution $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, where $\sigma_\varepsilon^2 = \sigma_{\varepsilon_{o,N}}^2 + \sigma_{\varepsilon_{s,N}}^2 \tau_{\max}^2$, $(t = \tau_{max})$. Imposing the upper-limit $\varepsilon_{\max}$ for the clock error via the probabilistic measure:

$$P_s = Pr\left(|\varepsilon| \geq \varepsilon_{\max}\right) = erfc\left(\frac{\varepsilon_{\max}}{\sqrt{2}\sigma_\varepsilon}\right),$$

where $erfc(x) \triangleq (2/\sqrt{\pi}) \cdot \int_x^\infty \exp\left(-t^2\right) dt$ and P_s denotes the synchronization error probability for pairwise synchronization. Thus, σ_ε can be determined when $\varepsilon_{\max}$ and the maximum allowable P_s are fixed. For instance, when P_s is limited to 0.1% and $\varepsilon_{\max}$ is 10ms, then the standard deviation of clock mismatch (σ_ε) has to be smaller than 3.04ms.

The maximum re-synchronization period with N beacons can be written as

$$\tau_{\max}^{(N)} = \sqrt{\frac{\sigma_\varepsilon^2 - \sigma_{\varepsilon_{o,N}}^2}{\sigma_{\varepsilon_{s,N}}^2}}\,. \tag{7.3}$$

Based on the lower bounds and asymptotic performance of the estimators, one can easily infer closed-form expressions of the variances $\varepsilon_{o,N}$ and $\varepsilon_{s,N}$ in terms of the variances $\varepsilon_{o,1}$ and $\varepsilon_{s,2}$, respectively. From the lower bound derived in Chapter V (see (5.5)), $\sigma_{\varepsilon_{o,N}}^2$ can be written with respect to N and $\sigma_{\varepsilon_{o,1}}^2$ as

$$\sigma_{\varepsilon_{o,N}}^2 = \frac{\sigma_{\varepsilon_{o,1}}^2}{N}.$$

Similarly, since the time differences between beacons are proportional to N and by far greater than the variance of delays, the following relationship can be obtained from (5.7):

$$\sigma_{\varepsilon_{s,N}}^2 = \frac{\sigma_{\varepsilon_{s,2}}^2}{(N-1)^2}, \quad N \geq 2.$$

Therefore, for $N \geq 2$, $\tau_{\max}^{(N)}$ can be rewritten as

$$\tau_{\max}^{(N)} = (N-1)\sqrt{\frac{\sigma_\varepsilon^2 - \frac{\sigma_{\varepsilon_{o,1}}^2}{N}}{\sigma_{\varepsilon_{s,2}}^2}}, \qquad N \geq 2. \tag{7.4}$$

Note that $\varepsilon_{s,1}$ can be obtained by the specifications of the crystal oscillator, and $\varepsilon_{o,1}$ and $\varepsilon_{s,2}$ can be determined by simple experimental tests. Therefore, the maximum re-synchronization period is proportional to the number of beacons, and performing clock skew estimation will significantly increase $\tau_{\max}^{(N)}$ since $\sigma_{\varepsilon_{s,1}} \gg \sigma_{\varepsilon_{s,2}}$.

Let us consider an example. Assume that the upper-limit for the clock error ϵ_{max} is 10 ms, the worst-case of sync error (ϵ_o) is 50 μs, and the worst-case of clock skew (ϵ_s) is 4.75 $\mu s/s$ as used in [14]. Then the maximum timing synchronization period τ_{max} can be obtained by

$$10\ m = 50\ \mu + 4.75\ \mu s/s \times \tau_{max}; \quad \tau_{max} \approx 35\ min.$$

If these bounds are satisfied within the range of 99.99 %, the set of standard deviations can be calculated as: $\sigma_\epsilon = 3.33\ m, \sigma_{\epsilon_o} = 16.67\ \mu$, and $\sigma_{\epsilon_s} = 1.58\ \mu$. Then, plugging these values into (7.3) gives $\tau_{max} \approx 35\ min$, which matches well with the above result. For $N \geq 2$, the maximum timing synchronization period $\tau_{max}^{(N)}$ becomes close to $35(N-1)\,(\sigma_{\varepsilon,1}/\sigma_{\varepsilon,2})\,min$ in this example.

3. Determination of Number of Beacons (N)

The goal of AMTS is to minimize the average number of message exchanges ($\overline{M}$). Hence, from (7.4), finding the optimal number of beacons (N) resume to solving the following optimization problem

$$\hat{N} = \arg\min_N \overline{M}, \tag{7.5}$$

with

$$\overline{M} = \frac{2BN}{\tau_{sync}^{(N)} + \tau_{\max}^{(N)}} = \begin{cases} \dfrac{2B}{\tau_{sync}^{(1)} + \sqrt{\dfrac{\sigma_{\varepsilon}^2 - \sigma_{\varepsilon_{o,1}}^2}{\sigma_{\varepsilon_{s,1}}^2}}} & N = 1 \\ \dfrac{2B}{\dfrac{\tau_{sync}^{(N)}}{N} + \dfrac{N-1}{N}\sqrt{\dfrac{\sigma_{\varepsilon}^2 - \dfrac{\sigma_{\varepsilon_{o,1}}^2}{N}}{\sigma_{\varepsilon_{s,2}}^2}}} & N \geq 2 \end{cases},$$

where $\tau_{sync}^{(N)}$ denotes the synchronization time with N beacons and will be estimated at the reference node for different Ns when the network is first established. Once N is estimated from (7.5), $\tau_{\max}^{(N)}$ can be obtained from (7.4). Fig. 26 illustrates the flowchart of the proposed synchronization scheme.

4. Sequential Multi-Hop Synchronization Algorithm

This section proposes the Sequential Multi-Hop Synchronization Algorithm (SMA) for energy-efficient timing message exchanges. The key idea of SMA is that both upper and lower level nodes are able to receive the same timing message simultaneously. SMA aims to reduce the number of timing message exchanges by using this property, which is proportional to the amount of energy consumption for synchronization. Fig. 27 compares the signaling strategy corresponding to the SMA (Fig. 27-(a)) with the conventional one (Fig. 27-(b)). The assumed network topology is a simple one-dimensional (linear) network with 4-levels. In TPSN, the clock synchronization must be done step-by-step as depicted in Fig. 27-(b).

In Fig. 27-(a), a node in the third level (*Node 3*) transmits a sync packet containing a time stamp (T_1) and a node identifier (ID) to a node in the second level (*Node 2*), then *Node 2* sends back an acknowledgement packet (ACK) to *Node 3* containing time stamps (T_1, T_2, and T_3) and its ID. Since a node in the first level (*Node 1*) also receives the ACK from *Node 2*, it sends back an ACK having time stamps (T_3, T_5, and T_6) and its ID to *Node 2*. Thus, *Node 1* regards the ACK from

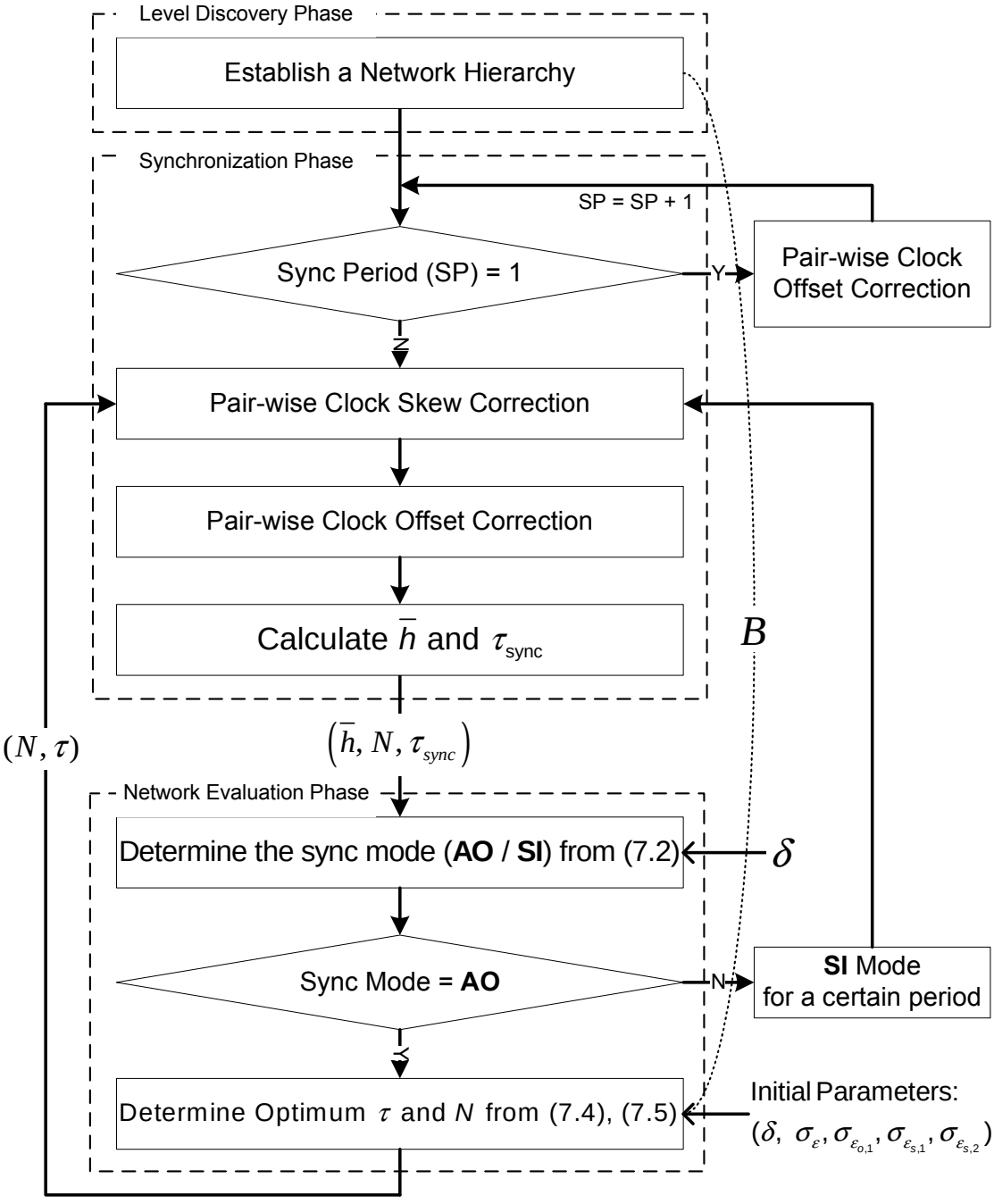

Fig. 26. Flowchart of AMTS.

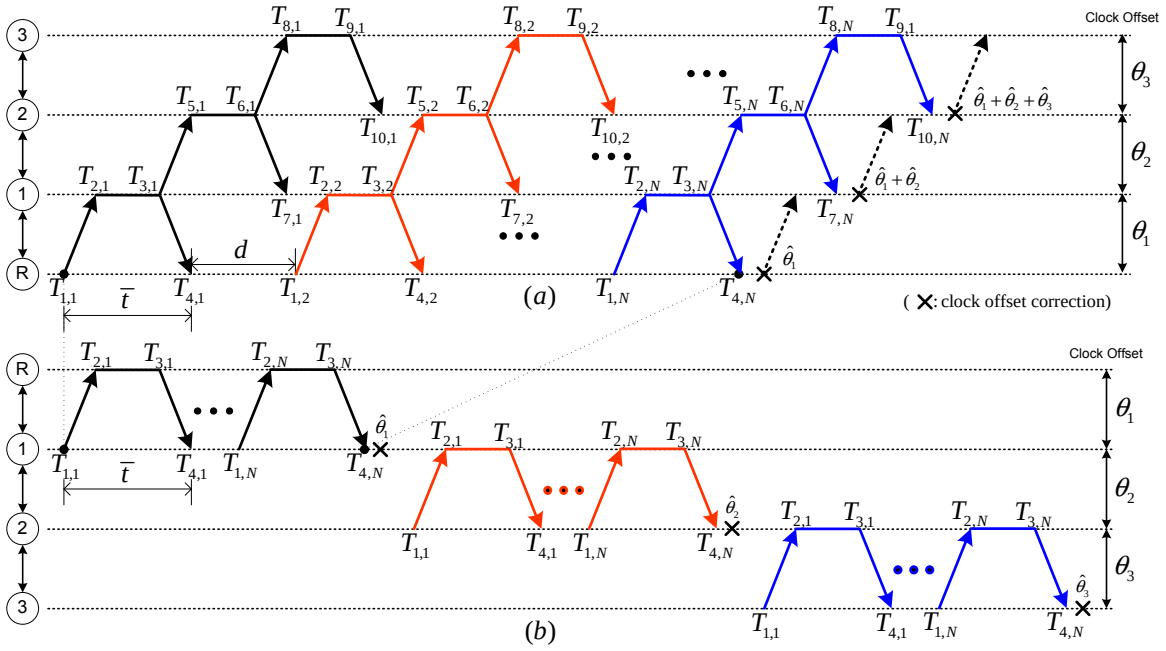

Fig. 27. Time synchronization models for multi-hop synchronization

Node 2 as a sync packet from *Node 2*. Similarly, the reference node (*Node R*) receives the ACK from *Node 1*, then returns an ACK to *Node 1*. This sequential procedure will be continued N times so as to increase the accuracy of clock estimation to an adequate level. Finally, *Node 1* estimates and adjusts clock offset, then sends it to *Node 2* to assist *Node 2* to synchronize to the reference. *Node 2* repeats the same procedure so that every node in the network can be synchronized. It can be easily extended to the networks having a large number of levels.

SMA presents two major advantages comparing with the signaling scheme adopted in TPSN: a decrement of the number of required timing messages for network-wide synchronization (M) by a factor of 2, and a significant reduction of the synchronization time ($\tau_{sync}^{(N)}$). In TPSN, the number of timing messages M is given by $M = 2BN$, while in SMA M is given by $M = (B+1)N + B$. Hence, the ratio between the

number of timing message exchanges required by TPSN and SMA is given by

$$R_m = \frac{2}{1 + \frac{1}{N} + \frac{1}{B}}. \tag{7.6}$$

Thus, SMA requires a lower number of timing messages for synchronization than TPSN when there are multiple branches and beacons in the network. Besides, as B and N are sufficiently large, R_m tends to 2.

From Fig. 27, the synchronization time can be written, respectively, as

$$\tau_{sync}^{(N)} \doteq N(\bar{t} + d) + B \cdot \bar{t} \qquad (SMA), \tag{7.7}$$

$$\tau_{sync}^{(N)} \doteq N(\bar{t} + d)B + (B - 1)d_L \qquad (TPSN), \tag{7.8}$$

where $\bar{t}$ stands for the average time for a timing message exchange (see Fig. 27), and d and d_L stand for the average delays between beacons ($d \ll d_L$). Note that, from (7.7) and (7.8), the synchronization time $\tau_{sync}^{(N)}$ is smaller than $N\tau_{sync}^{(1)}$, and $\tau_{sync}^{(N)}/N$ is monotonically decreasing as N increases. The ratio between the synchronization for TPSN and SMA is

$$R_t = \frac{N(\bar{t} + d)B + (B - 1)d_L}{N(\bar{t} + d) + B \cdot \bar{t}} = \frac{B\left(1 + \frac{d_L}{N(\bar{t}+d)} - \frac{d_L}{BN(\bar{t}+d)}\right)}{\left(1 + \frac{B \cdot \bar{t}}{N(\bar{t}+d)}\right)}. \tag{7.9}$$

Thus, if N is sufficiently large, R_t tends to B, which indicate a reduction of synchronization time by a factor of B. Besides, as the number of branches B is increasing, R_t goes to $N(1+d/\bar{t}+d/N\bar{t}) \approx N$, and therefore a reduction of the synchronization time by a factor of N. For arbitrary 2-dimensional networks, timing message transmission of the descendent nodes can be scheduled to avoid possible data collisions (which induce additional delays) using transmission scheduling schemes. Developing efficient signaling methods to minimize data collisions and delays in general 2-dimensional networks represents an interesting open research problem.

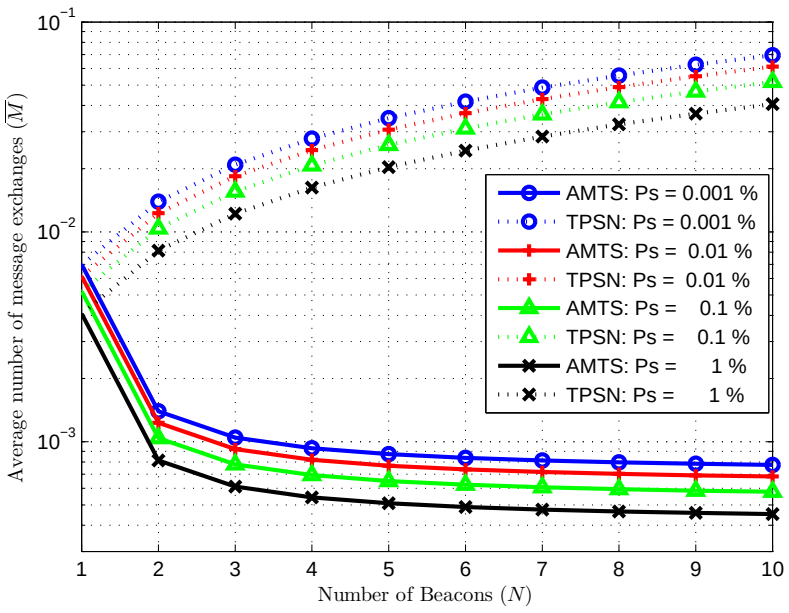

Fig. 28. Average number of message exchanges ($\overline{M}$) with respect to the number of beacons.

F. Simulation Results

Fig. 28 compares the performance of AMPS and TPSN in terms of the average number of message exchanges ($\overline{M}$) with respect to the number of beacons N when P_s is 1%, 0.1%, 0.01%, and 0.001%, respectively. This simulation is based on the linear network model where the depth of the network $B = 5$, $\epsilon_{max} = 10\ ms$, $\sigma_{\epsilon_o} = 16.67\ \mu$, $d = 10\ ms$, $\bar{t} = 400\ ms$, and $\sigma_{\epsilon_s} = 1.58\ \mu$.

It can be seen that AMTS requires a far less number of timing messages than TPSN when there exist multiple number of beacon transmissions. Moreover, the gap of the average number of required timing messages between AMTS and TPSN significantly increases as N increases, and thus AMTP is by far more energy-efficient than TPSN for large Ns. It can be also seen that a few number of beacons is enough to minimize $\overline{M}$ for AMTS. Besides, as expected, a larger number of beacons is required

to meet a more strict constraint on the network-wide error probability P_s. In practice, a lower N is highly preferable, since N is proportional to the synchronization time, i.e., a lower N induces better latency performance (although it may not be optimal in terms of energy consumption). In this simulation, the optimum values of N and $\tau^{(N)}$ will be obtained from the constraint $\tau^{(N)} \leq \tau_{sync}^{max}$. Experimental performance evaluation and performance comparison with other existing synchronization protocols represent future works.

G. Conclusions

A novel adaptive timing synchronization protocol for WSN has been proposed and analyzed in this study. Compared with TPSN, AMTS significantly reduces the overall network-wide energy consumption by employing key techniques in synchronization. First, AMTS adaptively determines the synchronization mode and adjusts the re-synchronization period with respect to the network status to minimize energy consumption. Second, it adapts the joint clock offset and skew estimators in order to achieve long term reliability of synchronization. The joint clock offset and skew synchronization has a significant benefit in terms of energy efficiency by increasing the re-synchronization period.

Consequently, combining these techniques helps to hugely reduce the number of required timing messages and increase the re-synchronization period, which induces highly energy-efficient timing synchronization. Moreover, the adaptive features in AMTS make it applicable to diverse types of wireless sensor networks with different requirements.

CHAPTER VIII

CONCLUSIONS AND FUTURE DIRECTIONS

In recent years, huge attention has been paid to WSNs due to their capability of serving a variety of purposes. Time synchronization is a significant part in WSNs, and a number of fundamental operations, like data fusion, power management and transmission scheduling, require accurate time synchronization. Since the conventional time synchronization protocol for the Internet can not be directly applied to WSNs, a number of synchronization protocols have been developed to meet the unique requirements of sensor network applications.

The importance of time synchronization also comes from the evolution of WSNs which has been driven by technological advances in diverse areas. For instance, unlike the currently deployed WSNs, next generation sensor networks may consist of dynamic mobile sensors or a mixture of static and dynamic sensors. In this scenario, far more sophisticated time synchronization protocols which efficiently deal with the mobility of sensors will be required. Indeed, as the network becomes more complicated, the role of time synchronization becomes much more important.

In this dissertation, basic features and theoretical backgrounds of the time synchronization problem in WSNs were first introduced and three basic approaches were analyzed and compared to reveal the general ideas and features of time synchronization protocols in WSNs. Besides, a survey of existing time synchronization protocols in the literature was provided including the most recent results.

As the main contributions of this dissertation, the problem of time synchronization has been studied in three different aspects targeting energy-efficient time synchronization in WSNs. First, a family of novel joint clock offset and skew estimators based on the classical two-way message exchange model has been developed. Second,

this dissertation proposed a new energy-efficient time synchronization protocol, called the Pairwise Broadcast Synchronization (PBS), which requires a far lesser number of timing messages (energy consumption) than other well-known ones and incurs no loss in synchronization accuracy. Finally, we proposed the Adaptive Multi-hop Timing Synchronization (AMTS) for the purpose of minimizing the overall network-wide energy consumption required for global synchronization based on the sender-receiver synchronization approach. The proposed synchronization schemes and theoretical analysis in this dissertation will be useful to develop (or select) more powerful synchronization protocols tailored specifically to the needs of particular sensor network applications.

A number of open research problems might be worth to investigate. Experimental performance evaluation and comparisons with other existing synchronization protocols represent a major open research work. More general random delay models might be needed for some sensor network applications. For instance, the Gamma distributed delay model might be a better choice than the exponential delay model in some cases due to its superior precision with the help of an extra free parameter. Moreover, a variety of sophisticated statistical techniques, such as jackknife, bootstrap, and Gibbs sampling, could be applied to improve the performance of clock estimation. Finally, another extension of PBS which dose not depend on the level hierarchy represents an interesting future research problem. Indeed, the proposed multi-cluster extension of PBS requires a searching procedure of the network hierarchy, whose complexity is proportional to the scale of the network.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," *Cumputer Networks*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, CA: Elsevier Inc., 2004.

[3] N. Bulusu and S. Jha, *Wireless Sensor Networks: A Systems Perspective*, MA: Artech House, 2005.

[4] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51-58, May 2000.

[5] D. L. Mills, "Internet time synchronization: the netwrok time protocol," *IEEE Trans. on Commununications*, vol. 39, no. 10, pp. 1482-1493, Oct. 1991.

[6] D. L. Mills, "A brief history of NTP time: confessions of an Internet timekeeper," *ACM Computer Communications Review*, vol. 33, no. 2, pp. 9-22, Apr. 2003.

[7] B. Sundararaman *et al.*, "Clock synchronization for wireless sensor networks: a survey," *Ad-Hoc Networks* vol. 3, no. 3, pp. 281-323, Mar. 2005.

[8] J. Elson and K. Romer, "Wireless sensor networks: a new regime for time synchronization," in *Proc. of the first workshop on hot topics in networks*, Princeton, NJ, Oct. 2002.

[9] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE networks*, pp. 45-50, July 2004.

[10] K. Romer, P. Blum, L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks*: Algorithms and Architectures, I. Stojmenovic, Ed. New York: John Wiley & Sons, 2005.

[11] J. E. Elson, "Time synchronization in wireless sensor networks," Ph.D. dissertation, University of California Los Angeles, Los Angeles, CA, Apr. 2003.

[12] W. Su, "Overview of time synchronization issues in sensor networks," *Embedded Systems*, Edited by R. Zurawski, CRC Press, 2005.

[13] B. M. Sadler and A. Swami, "Synchronization in sensor networks: an overview," in *Proc. of Military Communications Conference 2006*, Washington DC, Oct. 2006.

[14] S. Ganeriwal, R. Kumar and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. of the SenSys 2003*, Los Angeles, CA, Nov. 2003.

[15] J. R. Vig, "Introduction to quatz frequency standards," *Army Research Laboratory Electronics and Power Sources Directorate Technical Report* SLCET-TR-92-1, Oct. 1992.

[16] D. A. Howe, D. W. Alan, and J. A. Barnes, "Properties of signal sources and measurements methods," in *Proc. of the 35th Annual Symposium on Frequency Control*, Philadelphia, Pennsylvania, May 1981.

[17] A. Ephremides, "Energy concerns in wireless networks," *IEEE Trans. on Wireless Communications*, vol. 9, no. 4, pp. 48-59, Aug. 2002.

[18] J. Van Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, pp. 11-19, Sep. 2003.

[19] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Trans. on Computers*, vol. 36, no. 8, pp. 933-939, Aug. 1987.

[20] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12-24, Nov./Dec. 2002.

[21] Mica2 and Mica2Dot: http://www.xbow.com/Products/Wireless_Sensor_Networks .htm

[22] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems 2004*, Baltimore, MD, ACM Press, pp. 39-49, Nov. 2004.

[23] S. Narasimhan and S. S. Kunniyur, "Effect of network parameters on delay in wireless ad-hoc netwoks," University of Pennsylvania Technical Report, June 2004.

[24] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Mieghem, "Analysis of end-to-end delay measurements in Internet," in *Proc. of the Passive and Active Measurements Workshop*, Fort Collins, Colorado, Mar. 2002.

[25] S. Moon, P. Skelley, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. of the IEEE INFOCOM Conference on Computer Communications*, New York, Mar. 1999.

[26] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[27] J. Elson, L. Girod, D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. of the Fifth Symposium on Operating System Design and Implementation*, Boston, MA, Dec. 2002.

[28] K.-L. Noh and E. Serpedin, "Pairwise broadcast synchronization for wireless sensor networks," in *Proc. of the IEEE International Workshop: From Theory To Practice in Wireless Sensor Networks*, Helsinki, Finland, June 2007.

[29] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd Edition, NY: McGraw-Hill, 1991.

[30] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd Edition Reading, MA: Addison-Wesley, 1993.

[31] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithm with time-out control over networks with exponentially symmetric delays," *IEEE Trans. on Communications*, vol. 50, no. 10, pp. 1652-1661, Oct. 2002.

[32] D. R. Jeske, "On the maximum likelihood estimation of clock offset," *IEEE Trans. on Commununications*, vol. 53, no. 1, pp. 53-54, Jan. 2005.

[33] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. of the 7th ACM Sigmetrics Conference*, Madison, WS, vol. 26, pp. 11-21, June 1998.

[34] V. Paxson, "On calibrating measurements of packet transit times," Lawrence Berkeley National Laboratory Technical Report LBNL-41535, March 1998. Available online ftp://ftp.ee.lbl.gov/papers/vp-clocks-sigmetrics98.ps.gz.

[35] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I. Estimation Theory*, NJ: Prentice Hall, 1993.

[36] M. L. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *Proc. of the IEEE WCNC 2003*, New Orleans, LA, Mar. 2003.

[37] M. Lemmon, J. Ganguly and L. Xia, "Model-based clock synchronization in networks with drifting clocks," in *Proc. of the 2000 Pacific Rim International Symposium on Dependable Computing*, Los Angeles, CA, pp. 177-185, Dec. 2000.

[38] B. M. Sadler, "Local and broadcast clock synchronization in a sensor node," *IEEE Signal Proc. Letters*, vol. 13, no. 1, pp. 9-12, Jan. 2006.

[39] I. Sari, E. Serpedin and B. Suter, "On the joint synchronization of clock offset and skew in RBS-protocol," in *Proc. of the Military Communications Conference 2006*, Washington DC, Oct. 2006.

[40] F. Cristian, "A probabilistic approach to distributed clock synchronization," in *Proc. of the 9th International Conference on Distributed Computing*, San Jose, CA, pp. 288-296, June 1989.

[41] K. Arvind, "Probabilistic clock synchronization in distributed systems," *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 5, pp. 474-487, May 1994.

[42] D. Dolev, J. Halpern, and H. R. Strong, "On the possibility and impossibility of achieving clock synchronization," in *Proc. of the 16th Annual ACM Symp. on Theory of Computing*, Washington DC, Apr.-May 1984.

[43] J. Lundelius and N. Lynch, "An upper and lower bound for clock synchronization," *Information and Control*, vol. 62, no. 2/3, pp. 190-204, Aug.-Sep. 1984.

[44] S. Biaz and J. L. Welch, "Closed form bounds for clock synchronization under simple uncertainty assumptions," IPL: *Information Processing Letters*, vol. 80,

no. 3, pp. 151-157, Nov. 2001.

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

[46] D. Allan, "Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 34, no. 6, pp. 647-654, Nov. 1987.

[47] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. on Computers*, vol. 55, no. 2, pp. 214-226, Feb. 2006.

[48] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE JSAC*, vol. 23, no. 5, pp. 1085-1099, May 2005.

[49] A.-S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Trans. on Information theory*, vol. 52, no. 6, pp. 2725-2748, June 2006.

[50] A.-S. Hu and S. D. Servetto, "Asymptotically optimal time synchronization in dense sensor networks," in *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, San Diego, CA, pp. 1-10, Sep. 2003.

[51] Z. Tian, X. Luo, and G. B. Giannakis, "Cross-layer sensor network synchronization," in *Proc. of the 38th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, vol. 1, pp. 1276-1280, Nov. 2004.

[52] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *Proc. of the SenSys 2005*, San Diego, CA, Nov. 2005.

[53] S. PalChaudhuri, A. K. Saha and D. B. Johnson, "Adaptive clock synchronization in sensor networks," in *Proc. of the IPSN 2004*, Berkeley, CA, Apr. 2004.

[54] A. Syed, J. Heidemann, "Time synchronization for high latency acoustic networks," *Available as Technical Report ISI-TR-2005-602*, Apr. 2005.

[55] B. Awerbuch and R. G. Gallager, "A new distributed algorithm to find breath first search trees," *IEEE Trans. on Information Theory*, vol. 33, no. 3, pp. 315-322, May 1987.

[56] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless, "Minimum power configuration in wireless sensor networks," in *Proc. of the ACM MobiHoc 2005*, Urbana-Champaign, IL, pp. 390-401, May 2005.

APPENDIX A

DERIVATION OF THE JOINT MLE OF CLOCK OFFSET AND SKEW

Plugging the expression of $\hat{\theta}_{\mathrm{o}}$ (5.12) into that of $\hat{\theta}_{\mathrm{s}}$ (5.14) gives

$$(1+\hat{\theta}_{\mathrm{s}}) = \frac{\sum_{i=1}^{N}(T_{2,i}+T_{3,i}) - 2N\hat{\theta}_{\mathrm{o}}}{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})} = \frac{\sum_{i=1}^{N}\left[(T_{2,i}-\hat{\theta}_{\mathrm{o}})^2 + (T_{3,i}-\hat{\theta}_{\mathrm{o}})^2\right]}{\sum_{i=1}^{N}\left[(T_{1,i}+d)(T_{2,i}-\hat{\theta}_{\mathrm{o}}) + (T_{4,i}-d)(T_{3,i}-\hat{\theta}_{\mathrm{o}})\right]}. \tag{A.1}$$

$$\hat{\theta}_{\mathrm{o}} = \frac{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\sum_{i=1}^{N}(T_{2,i}^2+T_{3,i}^2) - \sum_{i=1}^{N}(T_{2,i}+T_{3,i})Q}{\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\sum_{i=1}^{N}(T_{1,i}+T_{4,i}) - 2NQ},$$

where $Q \triangleq \sum_{i=1}^{N}(T_{1,i}T_{2,i} + T_{3,i}T_{4,i} + (T_{2,i}-T_{3,i})d)$. Plugging (5.15) into (5.12) yields the MLE of clock skew

$$\hat{\theta}_{\mathrm{s}} = \frac{-2N\left[\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\sum_{i=1}^{N}(T_{2,i}^2+T_{3,i}^2) - Q\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\right]}{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})\left[\sum_{i=1}^{N}(T_{2,i}+T_{3,i})\sum_{i=1}^{N}(T_{1,i}+T_{4,i}) - 2NQ\right]} + \frac{\sum_{i=1}^{N}(T_{2,i}+T_{3,i})}{\sum_{i=1}^{N}(T_{1,i}+T_{4,i})} - 1.$$

APPENDIX B

DERIVATION OF THE PROPOSED CLOCK SKEW ESTIMATOR

The proposed clock skew can be derived by minimizing the expression (5.23), which is given by

$$\hat{\theta}'_{\mathrm{s}} = \arg\min_{\theta'_{\mathrm{s}}} \sum_{i=1}^{2} K_i \left|\theta'_{\mathrm{s}} - \delta_{(i)}\right| = \arg\min_{\theta'_{\mathrm{s}}} h(\theta'_{\mathrm{s}}),$$

where $h(\theta'_{\mathrm{s}}) \triangleq \sum_{i=1}^{2} K_i \left|\theta'_{\mathrm{s}} - \delta_{(i)}\right|$. Now divide the region of order statistics $\left\{\delta_{(i)}\right\}_{i=1}^{2}$ into 3 different regions as in Fig. 29, then the function $h(\theta'_{\mathrm{s}})$ in the 1st region becomes

$$h(\theta'_{\mathrm{s}}) = -\sum_{i=1}^{2} K_i \theta'_{\mathrm{s}} + \sum_{i=1}^{2} K_i \delta_{(i)} \qquad \theta'_{\mathrm{s}} \leq \delta_{(1)} \ \text{(region } \mathbf{1}\text{)}.$$

Since K_i is always positive, the corresponding estimate $\hat{\theta}'_{\mathrm{s}}$ is given by

$$\hat{\theta}'_{\mathrm{s}} = \arg\min_{\theta'_{\mathrm{s}}} h(\theta'_{\mathrm{s}}) = \delta_{(1)} \qquad \text{(region } \mathbf{1}\text{)}.$$

Similarly, in the 2nd region, the function $h(\theta'_{\mathrm{s}})$ becomes

$$h(\theta'_{\mathrm{s}}) = (K_1 - K_2)\,\theta'_{\mathrm{s}} + \left(K_2\delta_{(2)} - K_1\delta_{(1)}\right) \qquad \delta_{(1)} < \theta'_{\mathrm{s}} \leq \delta_{(2)} \ \text{(region } \mathbf{2}\text{)}.$$

Hence the estimate $\hat{\theta}'_{\mathrm{s}}$ is given by

$$\hat{\theta}'_{\mathrm{s}} = \arg\min_{\theta'_{\mathrm{s}}} h(\theta'_{\mathrm{s}}) = \begin{cases} \delta_{(1)} & K_1 > K_2 \\ \delta_{(2)} & K_1 < K_2 \\ \text{any value} & K_1 = K_2 \end{cases} \qquad \delta_{(1)} < \theta'_{\mathrm{s}} \leq \delta_{(2)} \ \text{(region } \mathbf{2}\text{)}.$$

Finally, in the 3rd region, the function $h(\theta'_{\mathrm{s}})$ takes the form

$$h(\theta'_{\mathrm{s}}) = \sum_{i=1}^{2} K_i \theta'_{\mathrm{s}} - \sum_{i=1}^{2} K_i \delta_{(i)} \quad \delta_{(2)} < \theta'_{\mathrm{s}} \ \text{(region } \mathbf{3}\text{)}.$$

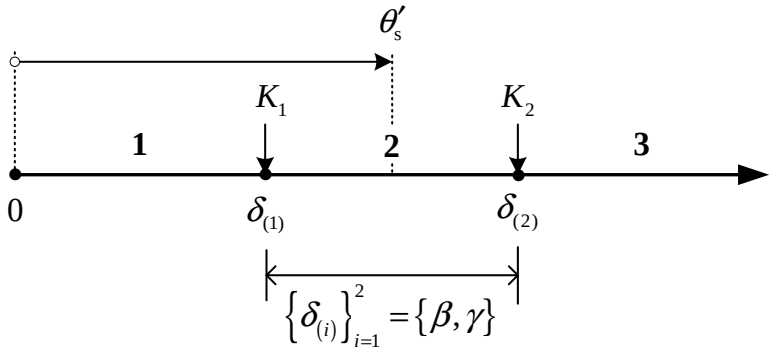

Fig. 29. Regions of the order statistics $\{\delta_{(i)}\}_{i=1}^{2}$.

So the estimate $\hat{\theta}'_s$ in this region is

$$\hat{\theta}'_s = \arg\min_{\theta'_s} h(\theta'_s) = \delta_{(2)} \qquad \text{(region } \mathbf{3}\text{)}\,.$$

Consequently, the estimate $\hat{\theta}'_s$ can be determined by choosing an adequate value between the order statistics $\{\delta_{(i)}\}_{i=1}^{2}$. The median of $\{\delta_{(i)}\}_{i=1}^{2}$ maximizes the likelihood function and minimizes the mean square error of the estimate. Therefore, the MLE of clock skew $\hat{\theta}_s$ for the exponential delay model is given by

$$\hat{\theta}_s = \frac{2}{\delta_{(1)} + \delta_{(2)}} - 1,$$

which is equivalent to the equation (5.24).

APPENDIX C

DERIVATION OF $F_Z(Z)$

Since $Z = X_{(1)} - Y_{(1)}$ and the order statistics $X_{(1)}$ and $Y_{(1)}$ are independent, $f_Z(z)$ can be found by transforming a joint distribution using the dummy variable $S = Y_{(1)}$. From the assumptions, the PDF of the uplink and downlink delays, X_i and Y_i, are given by

$$\begin{aligned} f_{X_i}(x) &= \frac{1}{\lambda_1} e^{-\frac{x}{\lambda_1}} \qquad x \geq 0, \\ f_{Y_i}(y) &= \frac{1}{\lambda_2} e^{-\frac{y}{\lambda_2}} \qquad y \geq 0. \end{aligned}$$

Using the result in [**?**, p. 9], the pdfs of the order statistics $X_{(1)}$ and $Y_{(1)}$ are given by

$$\begin{aligned} f_{X_{(1)}}(x) &= N\left(1 - F_{X_i}(x)\right)^{N-1} f_{X_i}(x) = \frac{N}{\lambda_1} e^{-\frac{N}{\lambda_1}x} \qquad x \geq 0, \\ f_{Y_{(1)}}(x) &= N\left(1 - F_{Y_i}(y)\right)^{N-1} f_{Y_i}(y) = \frac{N}{\lambda_2} e^{-\frac{N}{\lambda_2}y} \qquad y \geq 0. \end{aligned}$$

Since *Jacobian* of this transformation is 1, a joint distribution of RVs Z and S is given by

$$\begin{aligned} f_{Z,S}(z,s) &= f_{X_{(1)},Y_{(1)}}(z+s,s) = f_{X_{(1)}}(z+s) \cdot f_{Y_{(1)}}(s) \\ &= \frac{N^2}{\lambda_1\lambda_2} e^{-\frac{N}{\lambda_1}z} e^{-N\left(\frac{\lambda_1+\lambda_2}{\lambda_1\lambda_2}\right)s} \qquad z \geq -s,\ s \geq 0. \end{aligned} \tag{C.1}$$

Integrating (C.1) with respect to s yields

$$f_Z(z) = \begin{cases} \frac{N}{(\lambda_1+\lambda_2)} e^{-\frac{N}{\lambda_1}z} & z > 0 \\ \frac{N}{(\lambda_1+\lambda_2)} e^{\frac{N}{\lambda_2}z} & z < 0 \end{cases},$$

which is equivalent to the equation (5.2).

VITA

Kyoung Lae Noh was born in Seoul, Korea. He received the B.S. and M.S. degrees from the Department of Electronic Communications Engineering, Hanyang University, Seoul, Korea, in 2000 and 2002, respectively. From 1999 to 2002, he was a research assistant at the Communications Signal Processing Laboratory, Hanyang University. From 2002 to 2003, he was with the Platform R&D Center of SK Telecom, Seoul, Korea, where he managed the QoS of CDMA2000 EV-DO networks.

In 2003, he attended Texas A&M University to pursue his Ph.D. degree and has been a research assistant at the Wireless Communications Laboratory. In the summer of 2006, he was an Intern at the Digital Solution Center of Samsung Electronics, Seoul, Korea. His research interests are in the general areas of wireless communications with emphasis on wireless ad-hoc sensor networks. He received his Ph.D. degree in August 2007.

He can be reached at the following email address `abigbliss@gmail.com` .

The typist for this dissertation was Kyoung Lae Noh.