Signal Processing Techniques for Synchronization of Wireless Sensor Networks

Jaehan Lee[a], Yik-Chung Wu[b], Qasim Chaudhari[c] , Khalid Qaraqe[a], and Erchin Serpedin[*a]
[a]Texas A&M University, Electrical and Computer Eng., College Station, TX 77843-3128, USA;
[b]EEE Dept., University of Hong Kong,Pokfulam Road, Hong Kong;
[c] Iqra University, Islamabad, Pakistan;

ABSTRACT

Clock synchronization is a critical component in wireless sensor networks, as it provides a common time frame to different nodes. It supports functions such as fusing voice and video data from different sensor nodes, time-based channel sharing, and sleep wake-up scheduling, etc. Early studies on clock synchronization for wireless sensor networks mainly focus on protocol design. However, clock synchronization problem is inherently related to parameter estimation, and recently, studies of clock synchronization from the signal processing viewpoint started to emerge.

In this article, a survey of latest advances on clock synchronization is provided by adopting a signal processing viewpoint. We demonstrate that many existing and intuitive clock synchronization protocols can be interpreted by common statistical signal processing methods. Furthermore, the use of advanced signal processing techniques for deriving optimal clock synchronization algorithms under challenging scenarios will be illustrated.

Keywords: Clock Synchronization, Wireless Sensor Networks

1. INTRODUCTION

The clock or time synchronization problem in Wireless Sensor Networks (WSNs) resumes to finding a procedure for providing a common notion of time across the nodes of WSNs. In general, clock synchronization is viewed as a very critical factor in maintaining the good functioning of wireless sensor networks due mainly to their decentralized operation and timing uncertainties caused by the imperfections in hardware oscillators and message delays at the physical and Medium Access Control (MAC) layers.

The clock synchronization issue has been studied thoroughly in the areas of Internet and local-area networks (LANs) for the last several decades. Many existing synchronization algorithms rely on the clock information from GPS (Global Positioning System). However, GPS-based clock acquisition schemes exhibit some weaknesses: GPS is not available ubiquitously (for example, underwater, indoors, or under foliage) and requires a relatively high-power receiver, which is not feasible on tiny and cheap sensor nodes. This is the motivation for developing software-based approaches to achieve in-network time synchronization. Among many protocols that have been developed for maintaining synchronization in computer networks, NTP (Network Time Protocol) [1] is conspicuous due to its ubiquitous deployment, scalability, robustness related to failures, and self-configuration in large multi-hop networks. Furthermore, the combination of NTP and GPS has shown that it is capable of achieving high precision on the order of a few microseconds [2]. However, NTP is not proper for a wireless sensor environment, since wireless sensor networks pose a number of challenges of their own; to name a few, limited energy and bandwidth, limited hardware, latency, and unstable network conditions caused by mobility of sensors, dynamic topology, and multi-hopping. Therefore, clock synchronization protocols different from the conventional protocols are needed so as to cope with the challenges specific to WSNs.

One of the most popular clock synchronization protocols in wireless sensor networks is the Reference Broadcast Synchronization (RBS) [3], which is based on the post-facto receiver-receiver synchronization approach. In RBS, a reference broadcast message is sent by a node to two or more neighboring nodes which record their own local clocks at the reception of broadcasted message. After collecting a few readings, the nodes exchange their observations and a linear regression approach is used to estimate their relative clock offset and skew.

Dr. Serpedin* is the contact author. serpedin@ece.tamu.edu

Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies V,
edited by Paul Schiopu, George Caruntu, Proc. of SPIE Vol. 7821, 782102
© 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.881595

Timing Synch Protocol for Sensor Networks (TPSN) [4] is a conventional sender-receiver protocol which assumes two operational stages: the level discovery phase followed by the synchronization phase. During the level discovery phase, the wireless sensor network is organized in the form of a spanning tree, and the global synchronization is achieved by enabling each node to get synchronized with its parent (the node located in the adjacent upper level) by means of a two-way message exchange mechanism through adjusting only its clock offset.

Timing Synchronization protocol for High Latency acoustic networks (TSHL) [5] combines both of these approaches, namely the receiver-receiver (RBS) and sender-receiver (TPSN), in two stages. The Flooding Time Synchronization Protocol (FTSP) [6] also combines the two approaches in the sense that the beacon node sends its time stamps within the reference broadcast messages. The Time Diffusion Protocol (TDP) proposed by [7] achieves network-wide time equilibrium by using an iterative, weighted averaging technique based on a diffusion of messages involving all the nodes in the synchronization process. Also, the Asynchronous Diffusion Protocol (ADP) [8] uses a diffusion strategy similar to TDP, but the network nodes execute the protocol and correct their clocks asynchronously with respect to each other.

This paper is organized as follows. Section 1 yields a short introduction into the time synchronization problem, its history and importance. Section 2 discusses three general packet-based synchronization approaches for wireless sensor networks, focuses on the most representative synchronization protocols for wireless sensor networks by outlining their main features, and proposes a variety of statistical signal processing algorithms for improved estimation of clock phase offset. We also address the problem of building clock offset estimators that are robust to the distribution of network delays. Finally, Section 3 concludes the paper.

2. CLOCK SYNCHRONIZATION TECHNIQUES FOR WSNS

In [9], the synchronization protocols can be classified according to a variety of issues and characteristics. However, this paper deals with the synchronization protocols in terms of three general packet-based synchronization approaches; namely, sender-receiver synchronization, receiver-receiver synchronization, and receiver only synchronization. Basically, time synchronization in wireless sensor networks can be obtained 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 classified as sender-receiver synchronization (SRS) and receiver-receiver synchronization (RRS). Sender-receiver synchronization is based on the conventional model of two-way message exchanges between a pair of nodes. For receiver-receiver synchronization, the nodes to be synchronized receive a beacon packet from a common sender at first, and then compare their receiving time readings of the beacon packet to compute the relative clock offset. Most of the existing clock synchronization protocols count on one of these two mechanisms. For example, the Network Time Protocol (NTP) [1] and the Timing-sync Protocol for Sensor Networks (TPSN) [4] use sender-receiver synchronization because they depend on a series of pairwise synchronization assuming two-way timing message exchange scheme, whereas the Reference Broadcast Synchronization (RBS) [3] adopt receiver-receiver synchronization since it requires pairs of message exchanges among children nodes (except the reference) to compensate their relative clock offsets.

Recently, a new synchronization scheme, referred to as receiver-only synchronization (ROS) [10], was proposed. This approach aims at minimizing the number of required timing messages and energy consumption during synchronization while preserving a high level of precision. This method can be used to achieve network-wide synchronization with much fewer timing messages than other well-known existing protocols such as TPSN and RBS. In the following sections, we present and analyze each of three synchronization approaches mentioned above in more detail.

2.1 Sender-Receiver Synchronization (SRS)

As mentioned before, this approach is based on the traditional two-way timing message exchange mechanism between two adjacent nodes. The clock model for the two-way message exchange mechanism is shown in Fig. 1, where θ_A denotes the clock offset between Sensor A and Sensor B and timing messages are assumed to be exchanged multiple times. In this figure, the time stamps made during the i_{th} message exchange $T_{1,i}$ and $T_{4,i}$ are measured by the local clock of Sensor A, and $T_{2,i}$ and $T_{3,i}$ are measured by the local clock of Sensor B, respectively. Sensor A sends a synchronization packet, which includes the value of time stamp $T_{1,i}$ to Sensor B. Sensor B receives it at time $T_{2,i}$ and sends an acknowledgement packet to Sensor A at time $T_{3,i}$. This packet contains the value of time stamps $T_{1,i}$, $T_{2,i}$ and $T_{3,i}$. Finally, Sensor A receives the packet at time $T_{4,i}$.

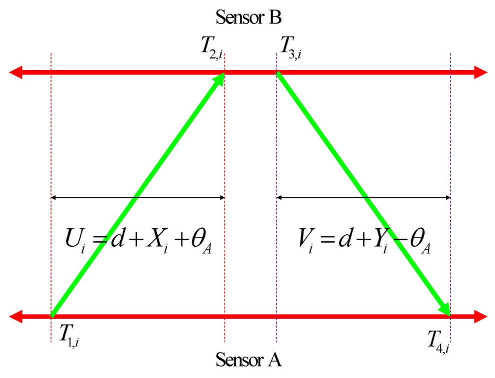

Fig. 1. Two-way message exchange model with clock offset (θ_A : clock offset, d : propagation delay, X_i, Y_i : random delays, $U_i = T_{2,i} - T_{1,i}$, $V_i = T_{4,i} - T_{3,i}$)

Packet delays can be decomposed into four distinct components: send time, access time, propagation time, and receive time [3]. First, *send time* is the time spent by the sender to construct the message, including kernel protocol processing and variable delays introduced by the operating system, e.g., context switches and system call overhead incurred by the synchronization application. Secondly, *access time* is the delay incurred while waiting for access to the transmission channel. Thirdly, *propagation time* is the time for the message to travel from the sender to the destination node through the channel since it left the sender. Lastly, *receive time* is the time for the network interface on the receiver side to get the message and notify the host of its arrival. These delay components can be divided into the fixed part d and the variable part X_i, Y_i. The variable portion depends on diverse network parameters such as network status and traffic. Therefore, any single delay model cannot satisfy every case. So far, several probability density function (PDF) models have been proposed for modeling the random delays, and Gaussian, exponential, Gamma, and Weibull PDFs are the most widely deployed [11], [12]. Herein, we mainly deal with the situations where random delays are Gaussian or exponential.

TPSN is one of the representative examples of sender-receiver synchronization and uses the two-way message exchange scheme shown in Fig. 1 to acquire the synchronization between two sensors [4]. In [4], the authors argue that for sensor networks, the classical approach of implementing a handshake between a pair of nodes is better than synchronizing a set of receivers. This observation results from time stamping the packets at the time when they are sent, namely, at MAC layer, which is indeed feasible for sensor networks. Reference [4] presented a comparison between the performance of TPSN with that of RBS based on the receiver-receiver synchronization approach, and illustrated that TPSN provides about two times better performance, in terms of accuracy, than RBS on the Berkeley motes platform. The TPSN protocol consists of two steps; namely, level discovery phase and synchronization phase. In level discovery phase, a hierarchical topology is created in the network. Each node is assigned a level in this hierarchical structure and a node belonging to level *i* can communicate with at least one node belonging to level *i* - 1. Only one node is assigned level 0, and it is referred to as the root node. Once the hierarchical tree structure is set up, the root node initiates the second step of the protocol, called the synchronization phase. In this second phase, a node with level *i* synchronize with a node with level *i* - 1. After all, every node is synchronized to the root node with level 0 and TPSN achieves network wide synchronization.

2.1.1 Clock Offset Estimation

Assuming that the clock frequencies of two sensors are equal during the synchronization period, and both X_i and Y_i are normal distributed random variables with mean μ and variance $\sigma^2/2$, U_i and V_i can be expressed as the following.

$$U_i = T_{2,i} - T_{1,i} = d + \theta_A + X_i, \quad V_i = T_{4,i} - T_{3,i} = d - \theta_A + Y_i \tag{1}$$

where the variables θ_A, d, X_i and Y_i denote the clock offset between the two nodes, the propagation delay, and the variable portions of delays, respectively. By doing some mathematical manipulations, we can obtain the likelihood function based on the observations U_i and V_i, which is given by

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

where N represents the number of message exchanges. Differentiating the log-likelihood function yields:

$$\frac{\partial \ln L(\theta_A)}{\partial \theta_A} = -\frac{2}{\sigma^2}\left[\sum_{i=1}^{N}(U_i - d - \theta_A - \mu) + \sum_{i=1}^{N}(V_i - d + \theta_A - \mu)\right]$$

(3)

Therefore, the maximum likelihood estimate (MLE) of clock offset is given by [15]

$$\hat{\theta}_A = \arg\max_{\theta_A}[\ln L(\theta_A)] = \frac{\sum_{i=1}^{N}(U_i - V_i)}{2N} = \frac{\bar{U} - \bar{V}}{2}. \tag{4}$$

The Cramer-Rao lower bound (CRLB) for the MLE can be derived by differentiating (4) with regard to θ_A:

$$\mathrm{var}(\hat{\theta}_A) \geq -E\left[\frac{\partial^2 \ln L(\theta_A)}{\partial \theta_A^2}\right]^{-1} = \frac{\sigma^2}{4N}. \tag{5}$$

For exponential delay model, the likelihood function based on the observations $\{U_i\}_{i=1}^{N}$ and $\{V_i\}_{i=1}^{N}$ is

$$L(\theta_A, \lambda) = \lambda^{-2N} e^{-\frac{1}{\lambda}\sum_{i=1}^{N}[U_i + V_i - 2d]} \cdot \prod_{i=1}^{N} I\left[U_i - \theta_A - d \geq 0, V_i + \theta_A - d \geq 0\right] \tag{6}$$

where λ is the mean value of X_i and Y_i, and $I(\cdot)$ represents the indicator function; in other words, $I(\cdot)$ is 1 whenever its inner condition holds, otherwise it is 0. In [16], it was proven that the maximum likelihood estimate of θ_A exists when d is unknown and is given by

$$\hat{\theta}_A = \frac{\min_{1\leq i\leq N} U_i - \min_{1\leq i\leq N} V_i}{2} \tag{7}$$

Note that when only one round of message exchange is performed (N = 1), the MLEs of clock offset for both Gaussian and exponential delay models become $\hat{\theta}_A = (U_1 - V_1)/2$, which is the same as the clock estimator given in [4].

By letting $\{U_{(i)}\}_{i=1}^{N}$ and $\{V_{(i)}\}_{i=1}^{N}$ represent the order statistics of the sequences of delay observations $\{U_i\}_{i=1}^{N}$ and $\{V_i\}_{i=1}^{N}$, respectively, equation (7) can be expressed as

$$\hat{\theta}_A = \frac{U_{(1)} - V_{(1)}}{2} = \theta_A + \frac{X_{(1)} - Y_{(1)}}{2}$$

(8)

where $X_{(1)}$ and $Y_{(1)}$ denote the order statistics corresponding to $\{X_i\}_{i=1}^{N}$ and $\{Y_i\}_{i=1}^{N}$, respectively. Let $Z = U_{(1)} - V_{(1)}$, then the probability density function of Z as a function of θ_A is expressed by

$$f_W(w;\theta_A) = \begin{cases} \dfrac{N}{(\lambda_1 + \lambda_2)} e^{-\frac{N}{\lambda_1}(w - 2\theta_A)}, & w > 2\theta_A \\ \dfrac{N}{(\lambda_1 + \lambda_2)} e^{\frac{N}{\lambda_2}(w - 2\theta_A)}, & w < 2\theta_A \end{cases} \tag{9}$$

where λ_1 and λ_2 are the mean values of X_i and Y_i, respectively. In case that the random delays are symmetric, $\lambda_1 = \lambda_2 = \lambda$, then the above equation becomes

$$f_W(w;\theta_A) = \frac{N}{2\lambda} e^{-\frac{N}{\lambda}|w - 2\theta_A|} \tag{10}$$

After some mathematical manipulations, we can obtain the CRLB of clock offset $\hat{\theta}_A$

$$\mathrm{var}(\hat{\theta}_A) \geq -E\left[\left(\frac{\partial \ln f_W(w;\theta_A)}{\partial \theta_A}\right)^2\right]^{-1} = \frac{\lambda^2}{4N^2} \tag{11}$$

It is clear that the performance of the maximum likelihood estimate of the clock offset strongly depend on the type of random delay models. In other words, observing the facts that the distribution of network delays in general can not be predicted accurately, and the estimators reported in the literature are not robust with respect to the unknown possibly time-varying distribution of network delays, we are facing the problem of building clock offset estimators that are robust to the distribution of network delays.

Now we are going to take a look at an estimator based on the Gaussian Mixture Kalman Particle Filter (GMKPF) [17] that turns out to be robust with respect to the distribution of random delays. With the observation sample $\mathbf{z}_k = [U_k, V_k]^T$, the goal is to find the minimum variance estimator of the unknown clock offset θ_A, where $U_k = T_{2,k} - T_{1,k} = d + \theta_A + L_k$ and $V_k = T_{4,k} - T_{3,k} = d - \theta_A + M_k$. For convenience, the notation $x_k = \theta_A$ will be used henceforth. Therefore, we need to determine the estimator, $\hat{x}_k = E\{x_k \mid \mathbf{Z}^l\}$, where $\mathbf{Z}^l$ denotes the set of observed samples up to time l, $\mathbf{Z}^l = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_l\}$. Since the clock offset value is assumed to be a constant, the clock offset can be modeled as obeying a Gauss-Markov dynamic channel model of the form, $x_k = Fx_{k-1} + v_{k-1}$, where F is the state transition matrix for clock offset. The additive noise component v_k can be modeled as Gaussian with zero mean and covariance $E\{v_k v_k^T\} = Q$. The vector observation model is expressed as

$$\mathbf{z}_k = \begin{bmatrix} d + \theta_A + L_k \\ d - \theta_A + M_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ -1 \end{bmatrix} x_k + \mathbf{n}_k$$

where the observation noise vector $\mathbf{n}_k = [L_k, M_k]^T$ accounts for the random delays. Note that the equations related to x_k and $\mathbf{z}_k$ recast our initial clock offset estimation problem into a Gauss-Markov estimation problem with unknown state.

Particle filtering is a sequential Monte Carlo sampling method built within the Bayesian paradigm. From a Bayesian viewpoint, the posterior distribution is the main entity of interest. However, due to the non-Gaussianity of the model, the analytical expression of the posterior distribution cannot be obtained in closed-form expression, except for some special cases like Gaussian or exponential probability density functions. Alternatively, particle filtering can be applied to approximate the posterior distribution by stochastic samples generated using a sequential importance sampling strategy.

Since the particle filtering with the prior importance function employs no information from observations in proposing new samples, its use is often ineffective and leads to poor filtering performance. Herein, we take a look at a slightly changed version of the Gaussian Mixture Sigma Point Particle Filter (GMSPPF) proposed in [18]. The GMSPPF is a family of methodologies that make use of hybrid sequential Monte Carlo simulation and a Gaussian sum filter to efficiently estimate posterior distributions of unknown states in a non-linear dynamic system. However, in our state space modeling, we do modify this method appropriately because of the linear model. Following [18], we will next describe briefly the general framework assumed by the GMKPF method, obtained by replacing the Sigma Point Kalman Filter (SPKF) with a Kalman Filter (KF). We next outline the main features of the approach. First, we remark that any probability density *p(x)* can be approximated as closely as desired by a Gaussian mixture model (GMM) of the following form [19], $p(x) \approx p_g(x) = \sum_{g=1}^{G} \alpha^{(g)} N(x; \mu^{(g)}, P^{(g)})$, where *G* represents the number of mixing components, $\alpha^{(g)}$ denotes the mixing weights and $N(x; \mu^{(g)}, P^{(g)})$ is a normal distribution. Hence, the predicted and updated Gaussian components, i.e., the means and covariances of the involved probability densities (posterior, importance, and so on) are calculated using the Kalman filter (KF) instead of the Sigma Point Kalman Filter (SPKF) [18], [20]. Since the state and observation equations are linear, the KF was employed instead of the SPKF. Therefore, the resulting approach is called the Gaussian mixture Kalman particle filter (GMKPF). In order to avoid the particle depletion problem in cases where the observation (measurement) likelihood is very peaked, the GMKPF represents the posterior density by a GMM which is recovered from the re-sampled equally weighted particle set using the Expectation-Maximization (EM) algorithm.

A pseudo-code for a GMKPF algorithm that is fit for estimating clock offsets in non-Gaussian delays is given below.

Algorithm

1) At time k-1, initialize the densities

- The posterior density is approximated by $p_g(x_{k-1} \mid \mathbf{z}_{k-1}) = \sum_{g=1}^{G} \alpha_{k-1}^{(g)} N(x_{k-1}; \mu_{k-1}^{(g)}, P_{k-1}^{(g)})$

- The process noise density is approximated by $p_g(v_{k-1}) = \sum_{i=1}^{I} \beta_{k-1}^{(i)} N(v_{k-1}; \mu_{v_{k-1}}^{(i)}, Q_{k-1}^{(i)})$
- The observation noise density is approximated by $p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(j)} N(\mathbf{n}_k; \mu_{\mathbf{n}_k}^{(j)}, R_k^{(j)})$

2) Pre-prediction step

- Calculate the pre-predictive state density using KF, $\tilde{p}_g(x_k \mid \mathbf{z}_{k-1})$
- Calculate the pre-posterior state density using KF, $\tilde{p}_g(x_k \mid \mathbf{z}_k)$

3) Prediction step

- Calculate the predictive state density using GMM, $\hat{p}_g(x_k \mid \mathbf{z}_{k-1})$
- Calculate the posterior state density using GMM, $\hat{p}_g(x_k \mid \mathbf{z}_k)$

4) Observation Update step

- Draw N samples $\{\chi_k^{(l)}; l = 1, \cdots, N\}$ from the importance density function, $q(x_k \mid \mathbf{z}_k) = \hat{p}_g(x_k \mid \mathbf{z}_k)$
- Calculate their corresponding importance weights: $\tilde{w}_k^{(l)} = \dfrac{p(\mathbf{z}_k \mid \chi_k^{(l)}) \hat{p}_g(\chi_k^{(l)} \mid \mathbf{z}_{k-1})}{\hat{p}_g(\chi_k^{(l)} \mid \mathbf{z}_k)}$
- Normalize the weights: $w_k^{(l)} = \tilde{w}_k^{(l)} / \sum_{l=1}^{N} \tilde{w}_k^{(l)}$
- Approximate the state posterior distribution using the EM-algorithm, $p_g(x_k \mid \mathbf{z}_k)$

5) Infer the conditional mean and covariance

- $\overline{x}_k = \sum_{l=1}^{N} \tilde{w}_k^{(l)} \chi_k^{(l)}$ and $\overline{P}_k = \sum_{l=1}^{N} \tilde{w}_k^{(l)} (\chi_k^{(l)} - \overline{x}_k)(\chi_k^{(l)} - \overline{x}_k)^T$

Figs. 2 and 3 show the MSE (Mean Square Error) of the estimators assuming that the random delay models are asymmetric Gaussian, exponential, Gamma, Weibull pdf, respectively. The subscripts 1 and 2 are used to differentiate the parameters of delay distributions corresponding to uplink and downlink, respectively. Note that the GMKPF (G=3 and G=5) performs much better when compared to SGML or SEML. The GMKPF-estimator did adaptively fit the posterior probability function(likelihood function) using EM, so the performance of GMKPF depend on the ability of GMM fitting for the distribution such as process noise, measurement noise, posterior PDFs, and so on.

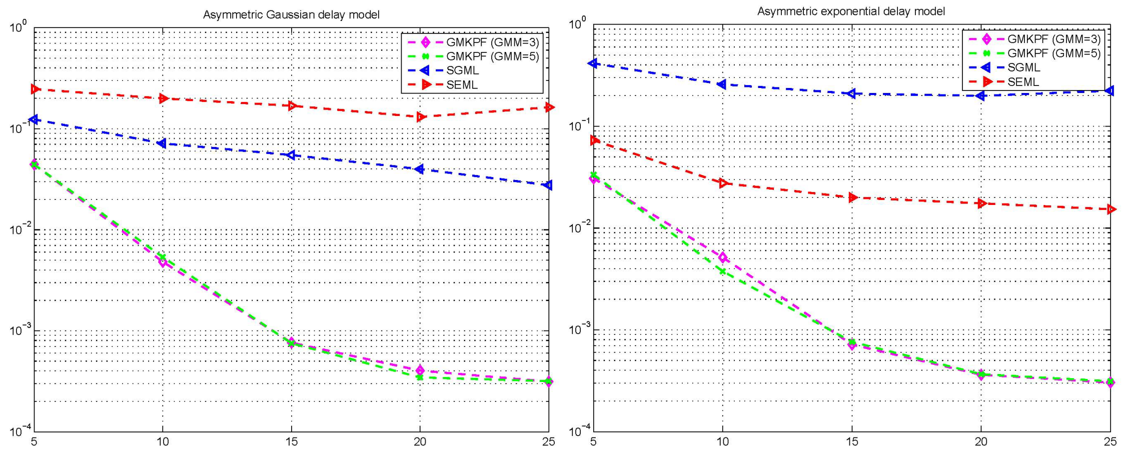

Fig. 2. MSEs of clock offset estimators for asymmetric Gaussian ($\sigma_1 = 1, \sigma_2 = 4$) and exponential random delays ($\lambda_1 = 1, \lambda_2 = 5$)

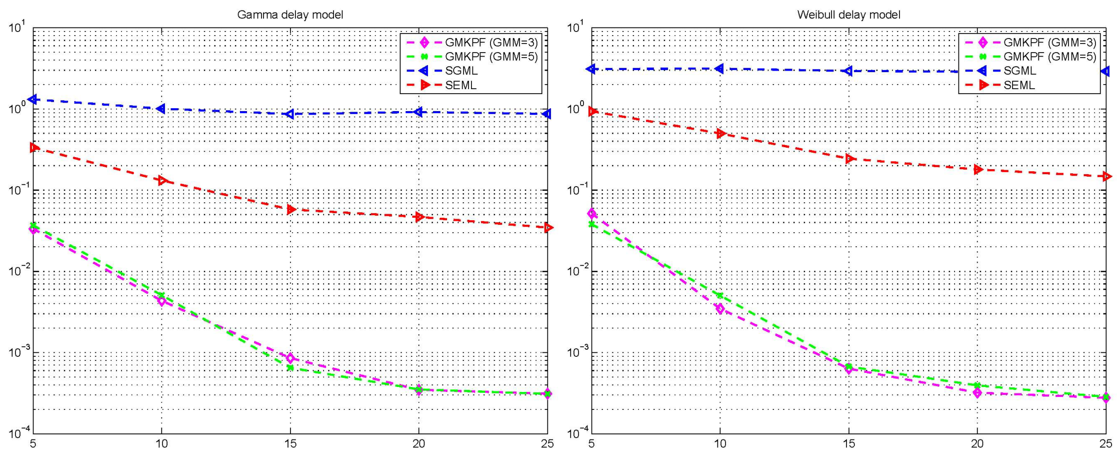

Fig. 3. MSEs of clock offset estimators for Gamma ($\alpha_1 = 2, \beta_1 = 1$) and Weibull random delays ($\alpha_1 = 2, \beta_1 = 2; \alpha_2 = 6, \beta_2 = 2$)

2.2 Receiver-Receiver Synchronization (RRS)

This method synchronizes a group of children nodes based on beacon messages from a parent node. RBS (Reference Broadcast Synchronization) protocol [3] is a representative example of receiver-receiver synchronization. The fundamental property of RBS is that a broadcast message is only used to synchronize a set of receivers with one another, in contrast with traditional protocols that synchronize the sender of a message with its receiver. By doing this, it eliminates the Send Time and Access Time from the critical path. This is a significant advantage for synchronization in a LAN, where the Send Time and Access Time are typically the main reasons for the non-determinism in the latency.

An RBS broadcast is always used as a relative time reference, and never to communicate an absolute time value. Indeed, this property removes the error caused by the Send Time and Access Time: each receiver synchronizes to a reference packet which was inserted into the physical channel at the same instant. The message itself does not include a timestamp generated by the sender, nor it is important exactly when it is sent. Actually, the broadcast does not even need to be a dedicated time synchronization packet. Almost any broadcast can be used to recover timing information – e.g., ARP packets in Ethernet, or the broadcast control traffic in wireless networks (e.g., RTS/CTS exchanges or route discovery packets). As mentioned above, RBS eliminates the effect of the error sources of Send Time and Access Time altogether; thus, the two remaining factors are Propagation Time and Receive Time. In [25], the Propagation Time is considered to be effectively 0. The propagation speed of electromagnetic signals through air is close to *c* (1nsec/foot), and through copper wire about 2*c*=3. In case of a LAN or ad-hoc network spanning tens of feet, propagation time is at most tens of nanoseconds, which does not contribute significantly to the sec-scale error budget. Moreover, RBS is only sensitive to the difference in propagation time between pair of receivers. Now let us take a look at one method of estimating clock offset and clock skew jointly in RBS protocol.

2.2.1 Joint Clock Offset and Clock Skew Estimation

Consider a parent node *P* and arbitrary nodes *A* and *B*, which are located within the communication range of the parent node *P*. In Fig. 4, we assume that both *Node A* and *Node B* receive the i_{th} beacon from *Node P* at $T_{2,i}^{(A)}$ and $T_{2,i}^{(B)}$ of their own 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 that $X_i^{(PA)}$ denotes the nondeterministic delay components (random portion of delays) and $d^{(PA)}$ denotes the deterministic delay component (propagation delay) from *Node P* to *Node A*, then $T_{2,i}^{(A)}$ can be written as

$$T_{2,i}^{(A)} = T_{1,i} + d^{(PA)} + X_i^{(PA)} + \theta_o^{(PA)} + \theta_s^{(PA)} \cdot (T_{1,i} - T_{1,1}) \tag{12}$$

$T_{1,i}$ stands for the transmission time at the reference node, $\theta_o^{(PA)}$ and $\theta_s^{(PA)}$ are the clock offset and skew of *Node A* with regard to the reference node, respectively. Likewise, the arrival time at *Node B* can be decomposed as

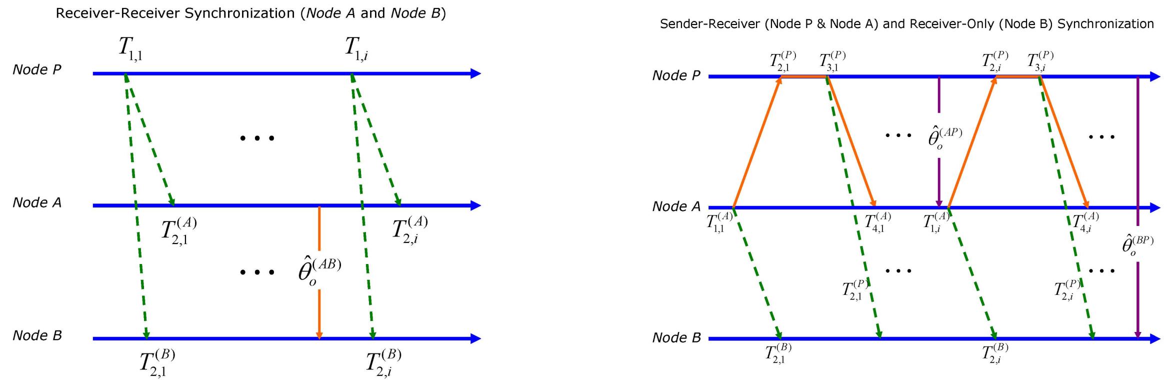

Fig. 4. Clock synchronization models in receiver-receiver and sender-receiver & receiver-only synchronizations

$$T_{2,i}^{(B)} = T_{1,i} + d^{(PB)} + X_i^{(PB)} + \theta_o^{(PB)} + \theta_s^{(PB)} \cdot (T_{1,i} - T_{1,1}) \tag{13}$$

where $d^{(PB)}$, $X_i^{(PB)}$, $\theta_o^{(PB)}$ and $\theta_s^{(PB)}$ represent the propagation delay, random delays, clock offset and skew of *Node B* with respect to the parent node, respectively. From these two equations,

$$T_{2,i}^{(A)} - T_{2,i}^{(B)} = \theta_o^{(BA)} + \theta_s^{(BA)} \cdot (T_{1,i} - T_{1,1}) + d^{(PA)} - d^{(PB)} + X_i^{(PA)} - X_i^{(PB)} \tag{14}$$

where $\theta_o^{(BA)} = \theta_o^{(PA)} - \theta_o^{(PB)}$ and $\theta_s^{(BA)} = \theta_s^{(PA)} - \theta_s^{(PB)}$ are thee relative clock offset and skew between *Node A* and *Node B* at the instant the i_{th} broadcast message packet is received from the reference node, respectively. Random delays $X_i^{(PA)}$ and $X_i^{(PB)}$ are assumed to be Gaussian random variables with mean μ and variance $\sigma^2/2$.

In equation (14), we can define the noise component as $z[i] = \mu' + X_i^{(PA)} - X_i^{(PB)}$, where $\mu' = d^{(PA)} - d^{(PB)}$ and $z[i]$ follows $N(\mu', \sigma^2)$. Putting $x[i] = T_{2,i}^{(A)} - T_{2,i}^{(B)} - \mu'$ and $w[i] = z[i] - \mu'$ gives the following matrix notation and the linear regression technique can be applied to the equation.

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w} \tag{15}$$

where $\mathbf{x} = [x[1]\ x[2] \cdots x[N]]^T$, $\mathbf{w} = [w[1]\ w[2] \cdots w[N]]^T$, $\boldsymbol{\theta} = [\theta_o^{(BA)}\ \theta_s^{(BA)}]^T$, and

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

By using the theorem from [14], [Theorem 3.2, pp. 44], we can obtain the minimum variance unbiased estimator (MVUE) for the relative clock offset and skew, which is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x} \quad , \quad \mathbf{I}(\boldsymbol{\theta}) = \frac{\mathbf{H}^T\mathbf{H}}{\sigma^2}$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the Fisher information matrix. By applying some mathematical operations, the joint clock offset and skew estimator can be written as [10]

$$\begin{bmatrix} \hat{\theta}_o^{(BA)} \\ \hat{\theta}_s^{(BA)} \end{bmatrix} = \frac{1}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \begin{bmatrix} \sum_{i=1}^{N} D_i^2 \sum_{i=1}^{N} x[i] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} \left[D_i \cdot x[i]\right] \\ N\sum_{i=1}^{N} \left[D_i \cdot x[i]\right] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} x[i] \end{bmatrix} \tag{16}$$

where $D_i = T_{1,i} - T_{1,1}$. Inverting the Fisher information matrix $\mathbf{I}(\boldsymbol{\theta})$ gives the Cramer-Rao Lower Bound (CRLB). Therefore,

$$\operatorname{var}(\hat{\theta}_o^{(BA)}) \geq \frac{\sigma^2 \sum_{i=1}^{N} D_i^2}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2}, \quad \operatorname{var}(\hat{\theta}_s^{(BA)}) \geq \frac{N\sigma^2}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \tag{17}$$

Also, under the exponential delay model for Reference Broadcast Synchronization (RBS), the joint maximum likelihood estimator (JMLE) and Gibbs sampler can be used for estimating the clock offset and skew [21], where lower and upper bounds for the mean squared errors (MSEs) of JMLE and Gibbs sampler are introduced in terms of the MSE of Uniform Minimum Variance Unbiased estimator (UMVUE) and Best Linear Unbiased Estimator (BLUE), respectively.

2.3 Receiver-Only Synchronization (ROS)

The communication range of a sensor is strictly limited to a circle whose radius is dependent on the transmission power owing to the energy constraint. In Fig. 3, *Node B* receives messages from both *Node P* and *Node A* under the assumption that *Node B* lie within the communication ranges of both *Node P* and *Node A*. Assuming that *Node P* is a reference node, and *Node P* and *Node A* perform a pairwise synchronization using two-way timing message exchanges [4], then all the nodes in the common coverage area of *Node P* and *Node A* can receive a series of synchronization messages containing the information related the time stamps of the pairwise synchronization. With this information, *Node B* can also be synchronized to the reference node without any extra timing message transmissions. This method is referred to as receiver-only synchronization (ROS). By using receiver-only synchronization, all the sensor nodes within the common coverage region of *Node P* and *Node A* can be synchronized by only receiving the timing messages.

2.3.1 Joint Clock Offset and Clock Skew Estimation

Let's assume that an arbitrary node, *Node B*, is located within the common coverage area of *Node P* and *Node A*. *Node B* can overhear the timing messages from the communication of *Node P* and *Node A*. In other words, *Node B* is able to observe a set of time stamps $\{T_{2,i}^{(B)}\}_{i=1}^{N}$ at its own local clock when it receives packets from Node A as shown in Fig. 3. Moreover, Node B can also obtain the information about a set of time stamps $\{T_{2,i}^{(P)}\}_{i=1}^{N}$ by receiving the packets transmitted by *Node P*. Considering the effects of both clock offset and skew, $T_{2,i}^{(P)}$ is represented by

$$T_{2,i}^{(P)} = T_{1,i}^{(A)} + d^{(AP)} + X_i^{(AP)} + \theta_o^{(AP)} + \theta_s^{(AP)} \cdot (T_{1,i}^{(A)} - T_{1,1}^{(A)}) \tag{18}$$

where $\theta_o^{(AP)}$ and $\theta_s^{(AP)}$ denote the relative clock offset and skew between *Node A* and *Node P*, respectively. Similarly, $T_{2,i}^{(B)}$ can be written as

$$T_{2,i}^{(B)} = T_{1,i}^{(A)} + d^{(AB)} + X_i^{(AB)} + \theta_o^{(AB)} + \theta_s^{(AB)} \cdot (T_{1,i}^{(A)} - T_{1,1}^{(A)}) \ . \tag{19}$$

The random delay portions $X_i^{(AP)}$ and $X_i^{(AB)}$ are assumed to be Gaussian random variables with mean μ and variance σ^2. From (16) and (17), we obtain

$$T_{2,i}^{(P)} - T_{2,i}^{(B)} = \theta_o^{(BP)} + \theta_s^{(BP)} \cdot (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AP)} - d^{(AB)} + X_i^{(AP)} - X_i^{(AB)} \tag{20}$$

where $\theta_o^{(BP)} = \theta_o^{(AP)} - \theta_o^{(BP)}$ and $\theta_s^{(BP)} = \theta_s^{(AP)} - \theta_s^{(BP)}$. (20) takes exactly the same form as (14). Therefore, the same mathematical operations can be applied to this equation so as to derive the joint clock offset and skew estimator for receiver-only synchronization scheme. Using similar steps as in RRS, we can obtain the minimum variance unbiased estimators (MVUEs) and the Cramer-Rao lower bounds (CRLBs) for the relative clock offset and skew in receiver-only synchronization, which is given by [10] and takes the same forms as (16) and (17), respectively. As a result, using these results, *Node B* can be synchronized to *Node A*. Similarly, all the other nodes in the common coverage area of *Node P* and *Node A* can be simultaneously synchronized to the reference node *Node P* without any extra timing messages, which enables a considerable amount of energy. Furthermore, there is no loss of synchronization precision in comparison with other approaches.

3. CONCLUSIONS

Recently, wireless sensor networks have attracted a tremendous attention in the literature. Clock synchronization plays a crucial role in a number of fundamental operations, such as power management, data fusion, and transmission scheduling. This paper provided an introduction to the clock synchronization problem of wireless sensor networks from a viewpoint of statistical signal processing.

REFERENCES

[1] Mills, D. L., "Internet time synchronization: the network time protocol," IEEE Transactions on Communications, 10, 1482-1493(1991).

[2] Bulusu, N., Jha, S., [Wireless Sensor Networks: A Systems Perspective], Artech House, 2005.

[3] Elson, J., Girod, L., and Estrin, D., "Fine-grained network time synchronization using reference broadcasts," Proc. 5th Symposium on Operating System Design and Implementation, 147-163 (2002).

[4] Ganeriwal, S., Kumar, R. and Srivastava, M. B. "Timing Synch Protocol for Sensor Networks," Proc. 1st International Conference on Embedded Network Sensor Systems, 138-149 (2003).

[5] Syed, A. and Heidemann, J., "Time synchronization for high latency acoustic networks," Proc. IEEE Infocom, 112 (2006).

[6] Maroti, M., Kusy, B., Simon, G. and Ledeczi, A., "The flooding time synchronization protocol," Proc. 2nd International Conference on Embedded Networked Sensor Systems, 39-49 (2004).

[7] Su, W. and Akyildiz, I. F., "Time-diffusion synchronization protocol for wireless sensor networks," IEEE/ACM Transactions on Networking, 13, 384397 (2005).

[8] Li, Q. and Rus, D., "Global clock synchronization in sensor networks," IEEE Transactions on Computers, 55, 214-226 (2006).

[9] Sundararaman, B., Buy, U. and Kshemkalyani, A. D., "Clock synchronization for wireless sensor networks: a survey," Ad-Hoc Networks, 3(3), 281-323 (2005).

[10] Noh, K., Serpedin, E. and Qaraqe, K., "A New Approach for Time Synchronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization," IEEE Transactions on Wireless Communications, 7(9), 3218-3322 (2008)

[11] Papoulis, A., [Probability, Random Variables and Stochastic Processes], McGraw-Hill, 1991.

[12] Leon-Garcia, A., [Probability and Random Processes for Electrical Engineering], Addison-Wesley, 1993.

[13] Abdel-Ghaffar, H. S., "Analysis of synchronization algorithm with time-out control over networks with exponentially symmetric delays," IEEE Transactions on Communications, 50, 1652-1661 (2002).

[14] Kay, S. M., [Fundamentals of Statistical Signal Processing, Vol. I. Estimation Theory], Prentice Hall, 1993.

[15] Noh, K., Chaudhari, Q, Serpedin, E. and Suter, B., "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," IEEE Trans. Commun., 55(4), 766-777 (2007).

[16] Jeske, D. R., "On the Maximum Likelihood Estimation of Clock Offset," IEEE Trans. Commun., 53(1), 53-54 (2005).

[17] Kim, J-S, Lee, J., Serpedin, E. and Qaraqe, K., "A Robust Estimation Scheme for Clock Phase Offsets in Wireless Sensor Networks in the Presence of non-Gaussian Random Delays", Signal Processing , 89(6), 1155-1161 (2009).

[18] Van der Merwe, R. and Wan, E., "Gaussian Mixture Sigma-Point Particle Filters for Sequential Probabilistic Inference in Dynamic State-Space Models," Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2003.

[19] Anderson, B. D. and Moore, J. B., [Optimal Filtering], Prentice-Hall, 1979.

[20] Julier, S. J. and Uhlmann, J. K., "A General Method for Approximating Nonlinear Transformations of Probability Distributions", RRG, Dept. of Engineering Science, University of Oxford, Nov. 1996.

[21] Sari, I., Serpedin, E., Noh, K-L, Chaudhari, Q. and Suter, B., "On the Joint Synchronization of Clock Offset and Skew in RBS-Protocol," IEEE Trans. Commun., 56(5), 700-703 (2008).