A Novel Interval-Based Protocol for Time Coordination in Wireless Sensor and IoT Networks – An Analytical Analysis

K. L. V. Sai Prakash Sakuru† and **N. Bheema Rao**, Non-members

ABSTRACT

This paper proposes a novel interval intersection-based protocol for time coordination in wireless sensor and IoT networks. The common notion of time amid the nodes in a distributed environment can be achieved through the message exchange process, which experiences random delay (send, access, propagation, and reception), thus making the time coordination process difficult. Several researchers have proposed algorithms to handle the error in estimation using various methods. This paper analytically analyzes the proposed novel unidirectional interval intersection method for mitigating the uncertainty in the interval width. The offset and slope estimation errors are then reduced under different conditions to verify the effectiveness of the proposed coordination algorithm. The model is simulated under three different delay models: uniform, normal, and truncated exponential. Their performance is then compared in terms of coordination efficiency.

Keywords: Time Coordination, Interval Intersection, Confined Delay, Expectation, Variance

1. INTRODUCTION

The job of time coordination protocols is to ensure all the clocks have the same notion of time. In the case of wireless sensor and IoT networks, time coordination in a distributed environment is critical and demanded by most wireless sensor applications. It is also one of the most popular research areas. Furthermore, many applications require time coordination for integrating the data generated by the sensor nodes, along with coordination among the sensor nodes for sleep/wake-up and energy optimization, teamwork for collision-free network access, time-critical industrial IoT [1, 2], and significantly, for source and target localization [3]. Therefore, the goal of the coordination protocol is to devise a mechanism to roll out the notion of universal time between all the nodes rather than correcting the clocks.

Based on a clock mechanism, the GPS global positioning system provides good time coordination. However, it has certain disadvantages, such as high energy consumption and cost per node since most nodes rely on batteries and need a clear sky. Furthermore, the system may not provide accurate results in underwater, indoor, and foliage environments. To address these issues, this study involves software-based algorithms and the use of wireless media to exchange the time information among sensor nodes.

In this paper, the related work carried out in the area of time coordination is explained in Section 2; the clock model and proposed algorithm for interval-based time coordination protocol and its analytical analysis in Section 3; the simulation results and reasoning in Section 4; and finally concludes with the findings in Section 5.

2. PREVIOUS WORK

Time coordination is an essential requirement for the efficient and effective functioning of time-sensitive wireless sensor applications and IoT networks. Sensor nodes are subjected to external parameters such as moisture and temperature, and the interference of these external parameters plays a direct role in the clock crystal performance, causing nodes to have a skewed notion of time [4]. Time offset and skew are the two main components informing this time inequality. The time offset is the time difference between the clocks, while skew is the difference in the rate of the running clocks. In a distributed environment, it is critical for these sensor nodes to coordinate their clocks together to perform any meaningful action. Time coordination protocols provide a mechanism for these sensor nodes to learn a global time across the entire network.

Several academicians and researchers have extensively studied time coordination protocols in wireless sensors and IoT networks. A well-established pattern among these protocols is to synchronize the node clocks through the exchange of time-stamped messages. However, such exchanges are subject to random delays at various stages like send, access, propagation, and reception [3, 5–8]. Most protocols try to handle these delays intelligently by increasing the frequency of these messages, thereby keeping the coordination error to a minimum. However, in most cases, nodes are driven by a

Manuscript received on July 26, 2021; revised on November 5, 2021; accepted on November 11, 2021. This paper was recommended by Associate Editor Kampol Woradit.

The authors are with the Department of Electronics & Communication Engineering, National Institute of Technology, Warangal, India.

†Corresponding author: sai@nitw.ac.in

Digital Object Identifier: 10.37936/ecti-eec.2022202.246901

battery, and the increased frequency of message communication and algorithmic computational load would drain more energy resources and negatively affect the overall node performance [9].

Time coordination design can be broadly classified into two approaches: 1) cluster-based and 2) hierarchical. Notable protocols such as TPSN [10] and FTSP [11] demand a hierarchical spanning tree to run coordination, whereas RBS [12], CB-TSP [13], CB-MCTS [14] use a cluster-based approach. In both cases, whenever a node fails, a considerable overhead is required to reconstruct the spanning tree/cluster.

Distributed time coordination protocols such as [15–17] use continuous-time representation, where nodes communicate only with their neighbors to achieve coordination. The distributed network does not demand any global topology knowledge. Instead, it only uses asynchronous broadcasts at each node.

In [18–21], using the bounded time interval between the transmission and reception of message packets and the random delay, the impact of clock drift was studied to build interval-based time coordination schemes rather than time instant estimates. These interval-based schemes use a bidirectional approach to estimate the offset, except for [21], where a unidirectional approach is employed. However, the drift was not estimated in all cases and compensated, which constrained the improvement of coordination quality.

In some other works, coordination protocols were built upon Wi-Fi beacons [22], Bluetooth beacons [23], and electromagnetic energy [24] to synchronize time across the entire network. While these gained much esteem for their novel approaches, they also incurred more coordination errors.

This paper proposes a novel unidirectional interval-based time coordination protocol and analytically analyzes it under the following assumptions: 1) all the sensor nodes need not start the coordination process at the same time, 2) the sensor nodes know the time interval a priori, and 3) the sensor nodes stop the coordination process once they have reached the coordination threshold value. Its performance over different random delay distribution environments such as uniform, normal, and truncated exponential is investigated to determine how the coordination precision is affected by its frequency.

3. INTERVAL-BASED TIME COORDINATION PROTOCOL

3.1 Terminology and System Model

This section presents a sender-receiver interval intersection-based time coordination protocol over confined delay paths, also referred to as master-slave unidirectional coordination. In this work, the following notations are used:

- $t_x(k)$ is the time at the sender node clock transmitting the k-th time-stamped packet;
- $r_x(k)$ is the time at which the receiver node clock receives the k-th time-stamped packet;

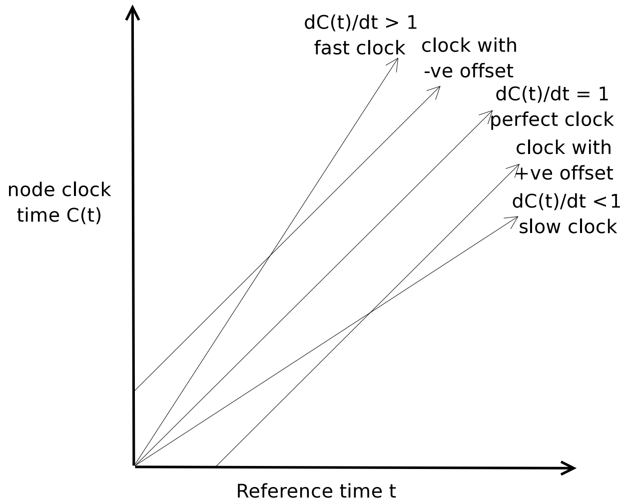

***Fig. 1**: Clock model.*

- The relative offset between the two clocks is represented by O_x;
- The relative skew between the two clocks is represented by S_x;
- The random delay induced by the k-th packet during the time-stamped message exchange is represented by $R(k)$;
- The confined delay interval is represented by $[C, D]$;
- The number of message packet exchanges is represented by k;
- $\underline{O}_x(k)$ is the left uncertainty limit of the interval of offset O_x after k message packet reception;
- $\overline{O}_x(k)$ is the right uncertainty limit of the interval of offset O_x after k message packet reception.

In a widespread environment, the sensor nodes clocks are coordinated through the exchange of message packets. These message packets might experience random send, access, propagation, and reception delays and are assumed to be confined within the limit $[C, D]$. One message packet exchange is acceptable if the end product demands clock accuracy exceeding the random delay incurred. Otherwise, more message exchanges are required to obtain the required clock accuracy. The clocks on the sensor nodes are assumed to be linear with progress monotonically increasing, which means that the clock moves forward but not backward. The relationship between the clocks on the sensor nodes can be modeled as shown in Fig. 1. The relationship between the sender-receiver pair clocks can be written as [21]:

$$r_x(k) = S_x * t_x(k) + O_x + R(k) \tag{1}$$

where the random delay incurred by the time-stamped packet transmitted from the sender to the receiver nodes is defined by R, and the k-th message packet transmission is represented by $R(k)$. It is assumed that $R(k) \in [C, D]$ and $0 \leq C \leq D < \infty$, i.e., $R(k)$ are confined delay random variables and C and D are known. By and large, the two variables skew S_x and offset O_x are adjustable to synchronize the clocks perfectly. Assuming all the clocks are running at the same speed, i.e., skew S_x is 1 and only differ by the offset, then Eq. (1) can be written as:

$$O_x = r_x(k) - t_x(k) - R(k). \tag{2}$$

If both the sender node and receiver node clocks are exactly the same without any offset, that is O_x is 0, and only the skew relative to each node is present, then it can be written as:

$$S_x = \frac{r_x(k) - R(k)}{t_x(k)}. \tag{3}$$

3.2 Known Slope S_x and Unknown Offset O_x

When the receiver node receives a time-stamped message packet from a sender node, it calculates the difference in the time taken between the message packet transmission time instant and receiving time instant. Assuming all the nodes' clocks are running at an identical speed, i.e., skew is 1, then the time difference is a combination of the offset and message delay time. Furthermore, since the delays $R(k)$ are assumed to be confined to limits $[C, D]$ is known to be a priori; from Eq. (2), the offset O_x should satisfy:

$$O_x \in \left[r_x(k) - t_x(k) - D,\ r_x(k) - t_x(k) - C\right], \quad \text{for } 1 \le k \le n. \tag{4}$$

Intersecting the above time intervals over n message transmissions leads to $O_x \in \left[\underline{O}_x(n),\ \overline{O}_x(n)\right]$ given by

$$\begin{aligned} \underline{O}_x(n) &= \max_{1 \le k \le n} \left(r_x(k) - t_x(k)\right) - D \\ &= O_x - \left(D - \max_{1 \le k \le n} R(k)\right), \end{aligned} \tag{5}$$

and

$$\begin{aligned} \overline{O}_x(n) &= \min_{1 \le k \le n} \left(r_x(k) - t_x(k)\right) - C \\ &= O_x + \left(\min_{1 \le k \le n} R(k) - C\right). \end{aligned} \tag{6}$$

Let

$$\widetilde{O}_x(n) := \left[\overline{O}_x(n) - \underline{O}_x(n)\right] \tag{7}$$

represent the width of unpredictability in the estimate of offset O_x after receiving n time-stamped packets, it can now be defined as

$$\widetilde{O}_x(n) = (D - C) - \left(\max_{1 \le k \le n} R(k) - \min_{1 \le k \le n} R(k)\right) \tag{8}$$

Note: no assumptions were made on the random sequence of delays $R(k)$ except that they are connected.

Let us define

$$G(n) = \max\left(R(n), R(n-1), \ldots, R(1)\right) \tag{9}$$

$$H(n) = \min\left(R(n), R(n-1), \ldots, R(1)\right) \tag{10}$$

then

$$\widetilde{O}_x(n) = D - C - \left(G(n) - H(n)\right). \tag{11}$$

Proposition 1: With probability one, (1) $G(n) \to D$, (2) $H(n) \to C$, and (3) $\widetilde{O}_x(n) \to 0$.

$(R(n), n \ge 1)$ is a sequence of real valued, independent and identically distributed random variables with confined support. R is a generic random variable distributed as $R(n)$. Let D denote the right end-point of the support of R and C be the left end point of the support R, i.e.,

$$D := \inf\{r : \mathrm{P}(R(1) > r) = 0\}$$
$$C := \sup\{r : \mathrm{P}(R(1) < r) = 0\}$$

Part (1): Define $S_n := \max_{1 \le i \le n} R(i)$, to show that $\lim_{n\to\infty} S_n = D$ almost certainly.

Proof: Almost certainly, S_n is a bounded, monotone non-decreasing sequence. This implies the random variable S can be defined such that $\lim_{n\to\infty} S_n = S$ almost certainly. To prove the proposition, it needs to be demonstrated that $\mathrm{P}(S = D) = 1$. Clearly, $\mathrm{P}(S \le D) = 1$, so it remains to show that $\mathrm{P}(S < D) = 0$.

For any $n \in \mathbb{N}$, $\varepsilon > 0$, $S < D - \varepsilon \Rightarrow S_n < D - \varepsilon$. Therefore,

$$\begin{aligned} \mathrm{P}(S < D - \varepsilon) &\le \mathrm{P}(S_n < D - \varepsilon) \\ &= \mathrm{P}(R < D - \varepsilon)^n. \end{aligned}$$

Since $\mathrm{P}(R < D - \epsilon) < 1$, taking $n \uparrow \infty$, we see that $\mathrm{P}(S < D - \epsilon) = 0$. Finally, since

$$\mathrm{P}(S < D) = \lim_{\epsilon \downarrow 0} \mathrm{P}(S < D - \varepsilon),$$

it can be observed that $\mathrm{P}(S < D) = 0$. This completes the proof. ■

Part (2): Define $S_n := \min_{1 \le i \le n} R(i)$, to show that $\lim_{n\to\infty} S_n = C$ almost certainly.

Proof: Almost certainly, S_n is a bounded, monotone non-increasing sequence. This implies that the random variable S can be defined such that $\lim_{n\to\infty} S_n = S$ almost certainly. To prove the proposition, it needs to be demonstrated that $\mathrm{P}(S = C) = 1$. Clearly, $\mathrm{P}(S \ge C) = 1$, so it remains to show that $\mathrm{P}(S > C) = 0$.

For any $n \in \mathbb{N}$, $\varepsilon > 0$, $S > C - \varepsilon \Rightarrow S_n > C - \varepsilon$. Therefore,

$$\begin{aligned} \mathrm{P}(S > C - \varepsilon) &\ge \mathrm{P}(S_n > C - \varepsilon) \\ &= \mathrm{P}(R > C - \varepsilon)^n \end{aligned}$$

Since $\mathrm{P}(R > C - \varepsilon) < 1$, taking $n \uparrow \infty$, it can be observed that $\mathrm{P}(S > C - \varepsilon) = 0$. Finally, since

$$\mathrm{P}(S > C) = \lim_{\varepsilon \downarrow 0} \mathrm{P}(S > C - \varepsilon),$$

it can be observed that $\mathrm{P}(S > C) = 0$. This completes the proof. ■

Thus, eventually, the offset can be estimated. To find the convergence rate of $\widetilde{O}_x(n)$, it is assumed that the $R(1), R(2), \ldots, R(i)$ are independent and identically distributed (i.i.d) random variables with a density function $f_R(r)$ and distribution $F_R(r)$ having lower and upper limits $[C, D]$. Therefore, the expectation of $\widetilde{O}_x(n)$ can be written as

$$\mathrm{E}[\widetilde{O}_x(n)] = (D - C) - \mathrm{E}[G(n) - H(n)] \tag{12}$$

Under this assumption, the joint pdf can be claimed as given by [25]

$$f_{GH}(g, h) = \begin{cases} n(n-1)(F(g) - F(h))^{n-2} f(g) f(h), & g \geq h \\ 0, & g \lneq h \end{cases}. \tag{13}$$

The expectation of $[G(n) - H(n)]$ is given by

$$\mathrm{E}[G(n) - H(n)] = \int_{g \geq h} (g-h) f_{GH}(g, h)\, dg\, dh \tag{14}$$

$$= n(n-1) \int_{g=C}^{D} \int_{h=C}^{g} (g-h)((F(g) - F(h))^{n-2} f(g) f(h)\, dg\, dh \tag{15}$$

The expectation and variance value of $\widetilde{O}_x(n)$ is given by

$$\mathrm{E}\left[\widetilde{O}_x(n)\right] = \int_C^D \left[(1 - F_R(g))^n + F_R(g)^n\right] dg \tag{16}$$

$$\begin{aligned} \mathrm{Var}\left[\widetilde{O}_x(n)\right] &= 2 \int_C^D \left[(g - C)(1 - F_R(g))^n + (D - g) F_R(g)^n\right] dg \\ &+ 2 \int_C^D \int_g^D \left[F_R(h) - F_R(g)\right]^n dh\, dg \\ &- \left(\int_C^D \left[(1 - F_R(g))^n + F_R(g)^n\right] dg\right)^2. \end{aligned} \tag{17}$$

3.3 Unknown Slope S_x and Known Offset O_x

When S_x is unknown and O_x is known, (e.g., if the two clocks are started by the same event) it can be analyzed in an analogous manner. From the equation $r_x(k) = S_x * t_x(k) + O_x + R(k)$, giving,

$$C \leq r_x(k) - S_x * t_x(k) - O_x \leq D.$$

This leads to

$$\frac{r_x(k) - D - O_x}{t_x(k)} \leq S \leq \frac{r_x(k) - C - O_x}{t_x(k)}$$

and the lower and upper bounds on S_x after n timestamps, $\underline{S_x(n)}$ and $\overline{S_x(n)}$ respectively, can be obtained as,

$$\begin{aligned} \underline{S_x(n)} &= \max_{1 \leq k \leq n} \left(\frac{r_x(k) - D - O_x}{t_x(k)}\right) \\ &= S_x + \max_{1 \leq k \leq n} \left(\frac{R(k) - D}{t_x(k)}\right) \end{aligned} \tag{18}$$

and

$$\begin{aligned} \overline{S_x(n)} &= \min_{1 \leq k \leq n} \left(\frac{r_x(k) - C - O_x}{t_x(k)}\right) \\ &= S_x + \min_{1 \leq k \leq n} \left(\frac{R(k) - C}{t_x(k)}\right). \end{aligned} \tag{19}$$

It should be noted that now, the choice of $t_x(k)$ determine $\underline{S_x(n)}$ and $\overline{S_x(n)}$. As before, let $\widetilde{S}_x(n) = \overline{S_x(n)} - \underline{S_x(n)}$

$$\begin{aligned} \widetilde{S}_x(n) &= \min_{1 \leq k \leq n} \left(\frac{r_x(k) - C - O_x}{t_x(k)}\right) - \max_{1 \leq k \leq n} \left(\frac{r_x(k) - D - O_x}{t_x(k)}\right) \\ &= \min_{1 \leq k \leq n} \left(\frac{R(k) - C}{t_x(k)}\right) - \max_{1 \leq k \leq n} \left(\frac{R(k) - D}{t_x(k)}\right). \end{aligned} \tag{20}$$

The analysis for the interval of S_x is carried out in a similar manner to O_x except that the random variables are independent but not identically distributed. Nevertheless, the following claim can be made.

Proposition 2: If $t_x(k)$ is a non decreasing sequence, (1) $\overline{S_x(n)} \to S_x$, (2) $\underline{S_x(n)} \to S_x$, and (3) $\widetilde{S}_x(n) \to 0$.

Proof: The proof is similar to that in Proposition 1 and is excluded.

Note that if $S_x \neq 1$, or if S_x is unknown, then in this analysis, the clock being used must be specified to characterize the delays $R(k)$. In all the above, it is assumed that $R(k)$ is measured using the clock of the send node.

3.4 Sender Node Uncertainty Interval

If the sender node has an uncertainty interval (t_x^l, t_x^r) and the time is incorrect, the receiver node receives the packet at its local time $r_x(n)$ and the packet contains the uncertainty interval. Let C and D be taken as a priori delay interval corresponding to the send node at the receive node. The overall delays with respect to time-stamped message transmissions and receptions are individualistic and can be written as

$$R(n) \in \left\{t_x^l + C,\ t_x^r + D\right\} \tag{21}$$

where $R(n)$ are independent and identically distributed across n and $R(1)$ has the probability density function $f_R(r)$ with limits $[C,\ D]$.

The uncertainty left and uncertainty right bounds $\underline{O}_x(n)$ and $\overline{O}_x(n)$ after n-th rounds of message transmissions is given by

$$\begin{aligned}\underline{O}_x(n) = O_x + t_x^l + C + R(n) \\ - \min\left(R(n), R(n-1), \ldots, R(1)\right),\end{aligned} \tag{22}$$

and

$$\begin{aligned}\overline{O}_x(n) = O_x + t_x^r + D + R(n) \\ - \max\left(R(n), R(n-1), \ldots, R(1)\right).\end{aligned} \tag{23}$$

The width of the uncertainty interval $\widetilde{O}(n)$ is given by

$$\widetilde{O}_x(n) = \overline{O}_x(n) - \underline{O}_x(n) \tag{24}$$

$$\begin{aligned}&= \left(t_x^r - t_x^l\right) + (D - C) \\ &- \Big(\max\left(R(n), R(n-1), \ldots, R(1)\right) \\ &- \min\left(R(n), R(n-1), \ldots, R(1)\right)\Big).\end{aligned} \tag{25}$$

The rest of the analysis is the same as Section 3.2, except for the uncertainty is present at the send node.

Proposition 3: With probability one, (1) $G(n) \to D$, (2) $H(n) \to C$, and (3) $\widetilde{O}_x(n) \to \left(t_x^r - t_x^l\right)$.

Proof: The proof is similar to that in Proposition 1 and is excluded.

3.5 Lossy Paths

One way to achieve confined delay is to drop the packets experiencing excessive delays. If a time-stamped packet is dropped, it is easy to see that Proposition 1 still holds. Let n transmissions be those received by the receiver node after k transmissions are made by the send node, and if a packet is dropped with the probability p, assuming that the $R(1), R(2), \ldots, R(i)$ are independent and identically distributed (i.i.d) binomial random variables and have a density function $f_R(r)$ and distribution $F_R(r)$ with lower and upper limits $[C,\ D]$, the expectation and variance value of $\widetilde{O}_x^l(n)$ is given by [25]

$$\mathrm{E}\left[\widetilde{O}_x^l(n)\right] = \sum_{n=0}^{n=k} \binom{k}{n} (1-p)^n p^{k-n} \int_C^D \left[(1 - F_R(x))^n + F_R(x)^n\right] dx \tag{26}$$

$$\begin{aligned}\mathrm{Var}\left[\widetilde{O}_x^l(n)\right] &= \sum_{n=0}^{n=k} \binom{k}{n} (1-p)^n p^{k-n} \\ &\Bigg(2\int_C^D \left[(x - C)(1 - F_R(x))^n + (D - x)F_R(x)^n\right] dx \\ &+ 2\int_C^D\int_x^D \left[F_R(y) - F_R(x)\right]^x dy dx \\ &- \left(\int_C^D \left[(1 - F_R(x))^k + F_R(x)^n\right] dx\right)^2\Bigg)\end{aligned} \tag{27}$$

where $\widetilde{O}_x^l(n)$ is the uncertainty in estimating offset with respect to the lossy path. There is a finite probability that the sensor may receive no transmission at all. In that case, Eqs. (26) and (27) are not valid.

3.6 Unknown Support

In the preceding analysis, it was assumed that the true support of $R(k)$ and the interval $(C,\ D)$, were known. That requirement can now be relaxed. In the system analysis, it will be assumed that $(C,\ D)$ in the preceding analysis is replaced by $\left(C_1,\ D_1\right)$. Four cases arise and each is analyzed.

If $C_1 \le C$ and $D_1 \ge D$ then it can be observed that $\underline{O_x(n)} \to D_1 - D$ and $\overline{O_x(n)} \to C - C_1$ and after a sufficient number of samples, the true support can be estimated. Thus, this case does not cause a problem. It is easy to see that $C_1 > C$ or $D_1 < D$ may pose some problems. To reiterate, $t_x(k)$ and $r_x(k)$ are the timestamps from the sender and the receiver. From Eq. (8), $\widetilde{O}_x(n) = (D_1 - C_1) - \big(\max_{1 \le k \le n}\left(r_x(k) - t_x(k)\right) - \min_{1 \le k \le n}\left(r_x(k) - t_x(k)\right)\big)$ are obtained. The event can be detected where $(D_1 < D) \cup (C_1 > C)$ if for $1 \le k \le (n - 1)$ on receipt of the n-th timestamp using the condition

$$\max_{1 \le k \le n-1}\left(r_x(k) - t_x(k)\right) - \min_{1 \le k \le n-1}\left(r_x(k) - t_x(k)\right) > D_1 - C_1 \tag{28}$$

In the above analysis, it is not possible to distinguish between $C_1 > C$ and $D_1 < D$. Since delays are non-negative, it can be assumed $C_1 = 0$. In the following, the case of $D_1 < n$ and fixing $C_1 = 0$ are analyzed. To simplify, it can be observed that Eq. (25) is the same as one of the following conditions being satisfied:

$$\left(r_x(n) - t_x(n)\right) - \min_{1 \le k \le n-1} \left(r_x(k) - t_x(k)\right) > D_1 \quad (29)$$

$$\max_{1 \le k \le n-1} \left(r_x(k) - t_x(k)\right) - \left(r_x(n) - t_x(n)\right) > D_1. \quad (30)$$

The probability of detecting that $D_1 < D$ for the first time after n timestamps will now be analyzed. Eqs. (29) and (30) are reduced to

$$R(n) - \min_{1 \le k \le n-1} R(k) > D_1 \quad (31)$$

$$\max_{1 \le k \le n-1} R(k) - R(n) > D_1. \quad (32)$$

Regarding the estimation error in D at the n-th stage, given it was not detected in the previous $n - 1$ stages, the equations can be used but they cannot be satisfied for the first $n-1$ timestamps. Therefore, the joint probability with the event is required

$$\max_{1 \le k \le n-1} R(k) - \min_{1 \le k \le n-1} R(k) < D_1 \quad (33)$$

Let us denote the event of detection of $D_1 < D$ from Eqs. (29) and (30) as Z_1 and Z_2, respectively, and the event of no detection $D_1 < D$ from Eq. (28) as Z_3. It can be observed that Z_1 and Z_2 are mutually exclusive, and the required probability of detection of $D_1 < D$ is given by

$$\mathrm{P}\left(Z_3 \cap (Z_1 \cup Z_2)\right) = \mathrm{P}\left(Z_3 \cap Z_1\right) + \mathrm{P}\left(Z_3 \cap Z_2\right). \quad (34)$$

Denoting $G = \min_{1 \le k \le n-1} R(k)$ and $H = \max_{1 \le k \le n-1} R(k)$, and the joint pdf given in Eq. (13), the equation for the probability of detection of error is obtained as,

$$\mathrm{P}\left(Z_3 \cap Z_1\right) = \int_C^D \int_g^{\min(g+D_1,\, D)} f_{GH}^{(n-1)}(g,h) \left(1 - F_R(g + D_1)\right) dg\,dh \quad (35)$$

$$\mathrm{P}\left(Z_3 \cap Z_2\right) = \int_C^D \int_g^{\min(g+D_1,\, D)} f_{GH}^{(n-1)}(g,h) F_R(h - D_1) dh\,dg \quad (36)$$

where $f_{GH}^{(n-1)}(g,h)$ is the density corresponding to the distribution $F_{GH}^{(n-1)}(g,h) = P(H - G < D_1)$.

3.7 Comparison Between the Proposed Algorithm and Existing Work

Most of the time, synchronization algorithms use regression techniques for offset and slope estimation [11, 12, 15, 18, 26]. When a time-stamped packet is exchanged between a node pair, the transmit and receive timestamps do not give the actual times since the clocks are not synchronized. The transmit and received timestamps are the linear function of skew and offset. The linear regression algorithm tries to fit a straight line among the points so that the Mean Square (MS) error is minimal. Each point influences the fitted line. Assuming the delay R is random, this affects the synchronization accuracy.

Let $\left(t_x(k),\, r_x(k)\right)$ $\forall\, k$, be the data points available and linearly related as

$$r_x(k) = S * t_x(k) + O + R(k) \quad (37)$$

where $t_x(k)$ is the k-th packet transmission timestamp of the sender node, $r_x(k)$ is the k-th packet reception timestamp of the receiver node, O is the offset between sender and receiver node clocks, S is the skew between sender and receiver node clocks, and $R(k)$ is the random delay incurred by the message packet.

The linear estimation problem (Linear MS Estimation) is the estimation of random variable R in terms of a linear function O of $t_x(k)$. Assuming the skew $S = 1$, i.e., both clocks at the receiver and sender are running at the same speed, the problem now is to find the offset O so as to minimize the MS error e.

$$e = \sum_{k=1}^{n} \left\{r_x(k) - t_x(k) - O - R(k)\right\}^2. \quad (38)$$

It is maintained that $e = e_m$ is minimal if $de/dO = 0$. Therefore,

$$-2\sum_{k=1}^{n} \left\{r_x(k) - t_x(k)\right\} + 2\sum_{k=1}^{n} \{O\} + 2\sum_{k=1}^{n} \{R(k)\} = 0. \quad (39)$$

The estimate of O is given by:

$$\hat{O} = \frac{\sum_{k=1}^{n} \left(r_x(k) - t_x(k)\right)}{n} \quad (40)$$

that is

$$\hat{O} = \frac{\sum_{k=1}^{n} (R(k) + O)}{n} \quad (41)$$

where the difference between the receive and send time stamps $r_x(k) - t_x(k)$ is equal to the offset and random delays incurred by the message packet corresponding to send, access, propagation, and receive, defined as the $R(k) + O$. The random delay $R(k)$ is assumed to be k-th packet transmission delay with support $[C,\, D]$. Therefore, it can be rewritten as

$$\hat{O} = \frac{\sum_{k=1}^{n} R(k)}{n} + \frac{\sum_{k=1}^{n} (O)}{n}. \quad (42)$$

***Table 1**: Simulation parameters*

Parameters	Values
Lower interval limit	0.2 units of time
Upper interval limit	0.7 units of time
Number of message packets	50, 100
Unidirectional packet exchange	every clock tick
Simulation runs	10 000

Thus,

$$e_m = \widehat{O} - O = \frac{\sum_{k=1}^{n} R(k)}{n}. \tag{43}$$

The expectation of e_m is:

$$\mathrm{E}[e_m] = \mathrm{E}\left[\frac{\sum_{k=1}^{n}(R(k))}{n}\right] = \mathrm{E}[R(k)] \tag{44}$$

The variance of e_m is:

$$\begin{aligned}\mathrm{Var}\left[e_m\right] &= \mathrm{Var}\left[\frac{\sum_{k=1}^{n}(R(k))}{n}\right] \\ &= \frac{1}{n^2}\mathrm{Var}\left[\sum_{k=1}^{n}(R(k))\right] \\ &= \frac{1}{n}\mathrm{Var}\left[R(k)\right]\end{aligned} \tag{45}$$

The point estimation uses one single number as the best estimator for offset O corresponding to the n data set received. Whereas interval estimation uses a range of numbers within which the parameter is believed to fall with some confidence value. Hence, the estimated values of the proposed model are more complete than linear regression.

4. SIMULATIONS AND RESULTS

In this section, the simulation environment is considered to endorse the efficacy of the interval intersection-based time coordination protocol over confined delay paths for wireless sensor and IoT networks. In these simulations, a sender-receiver node pair is considered with MATLAB used for implementation. For simulation purposes, at every time unit, the unidirectional message packet is exchanged, and the coordination process initiated at the receiver node as shown in Table 1. The coordination process occurs at every one clock tick and continues until it reaches the threshold value defined by the application.

The random delays experienced by the message packet are modeled as a uniformly distributed delay over the said

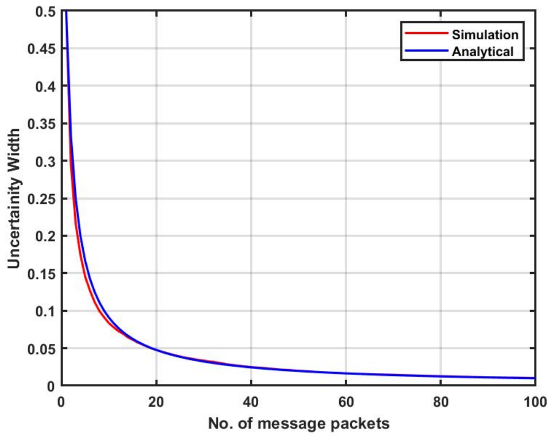

***Fig. 2**: Uncertainty width of the interval-based time coordination under uniform distribution.*

interval. The expectation and variance of the offset error is given by:

$$\mathrm{E}\left[\widetilde{O}_x(n)\right] = \frac{2*(D-C)}{n+1} \tag{46}$$

$$\mathrm{Var}\left[\widetilde{O}_x(n)\right] = \frac{2*(D-C)^2(n-1)}{(n+1)^2(n+2)} \tag{47}$$

Fig. 2 shows the simulation and analytical results of the interval-based time coordination protocol. The analytical analysis and algorithm simulations are as per the design. The uncertainty interval width converges as the number of message packets are exchanged between the node pair. From the graph, it can be inferred that after 20 message packets, the drop in uncertainty interval is 90%.

Fig. 3 shows the simulation of the slope and offset expectation error of the interval-based time coordination protocol under the uniform distribution random delay model. The uncertainty interval width of both slope and offset converges as the number of message packets are exchanged between the node pair, and slope converges faster than the offset. From the graph, it can be inferred that after 20 message packets, the slope can be estimated with more accuracy than the offset estimation.

Figs. 4 and 5 show the simulation of the offset expectation error of the interval-based time coordination protocol under the normal and truncated exponential distribution random delay models. The uncertainty interval width of the offset converges as the number of message packets are exchanged between the node pair in both cases. From the graph, it can be inferred that the interval-based time coordination protocol performs better in the case of the truncated exponential distributed random delay model than the normal and uniform distributed random delay models. The truncated exponential random delay model outperformed both the uniform random delay and the normal random delay model distribution.

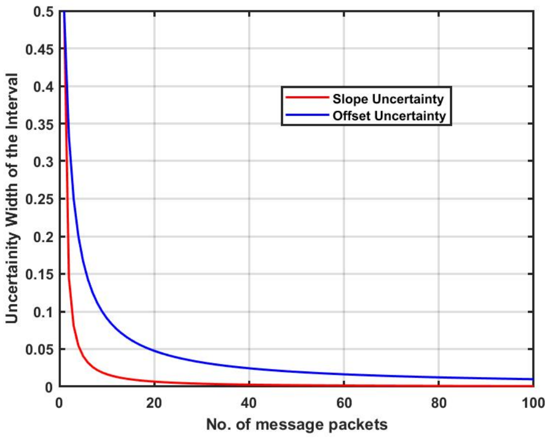

Fig. 3*: Uncertainty width of the interval-based time coordination with both offset and slope under a uniform distribution environment.*

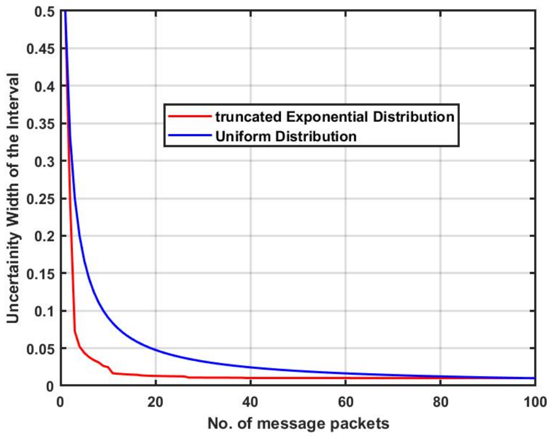

Fig. 5*: Uncertainty width of offset in the case of interval-based time coordination under uniform and truncated exponential distribution environments.*

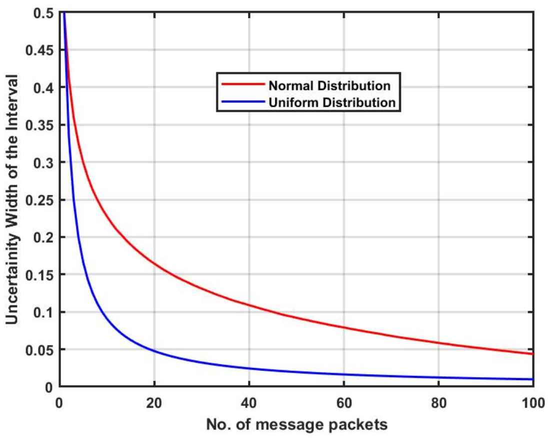

Fig. 4*: Uncertainty width of offset in the case of interval-based time coordination under normal and uniform distribution environments.*

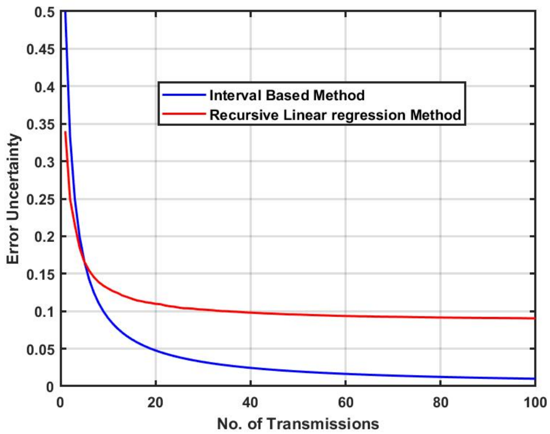

Fig. 6*: Uncertainty width in the case of the interval-based time coordination and linear regression model under a uniform distribution environment.*

Fig. 6 shows the simulation offset error uncertainty of the interval-based time coordination protocol and recursive linear regression protocol under the uniform distribution random delay model. In the case of linear regression protocols, the point estimation uses one single number as the best estimator for offset O corresponding to the n data set received. Whereas interval-based protocol estimation uses a range of numbers within which the parameter is believed to fall with some confidence value. Hence, the estimated values of the proposed model are more complete than linear regression.

5. CONCLUSION

This paper presents the unidirectional interval-based time synchronization protocol over confined delay paths and its complete analysis. The algorithm updates monotonically, diminishing new limits of the interval as and when it receives a new time-stamped message packet from the sender node. These updates reduce the requirement for a large memory on the sensor node. The algorithm was studied under various constraints, such as i) assuming all nodes know the interval bounds, ii) packets are not lost during transmission, iii) packets are lost during transmission (excess delay in transmission due to congestion on the network), and iv) if the node is not aware of the interval bounds. The performance of the interval-based time synchronization protocol is also studied against those using linear regression to estimate the offset. Simulations are carried out under different distribution environments such as uniform, Gaussian-normal, and truncated exponential to observe the algorithm's performance. From the simulations, it can be observed that after 20 message packets, the drop in uncertainty interval is 98% in the case of truncated exponential distribution, compared to 90% for uniform distribution and 68% for normal distribution. The performance of the protocols was also studied against linear regression protocols. The linear regression protocols provide point estimates for offset, whereas the interval-based protocol provides a range of numbers with some confidence value. A large number of message packets should be used to estimate the error for regression protocols rather than the interval-based protocol. The results show that as the number of time-stamped packets grows, the uncertainty interval reduces, and if the interval bounds are selected carefully, it is bound to converge, allowing offset and skew to be estimated.

APPENDIX: DETAILED ANALYSIS

On receiving the 1 time-stamped packet

$$r_x(1) \in \left[t_x(1) + O_x + C,\ t_x(1) + O_x + D\right]. \tag{48}$$

The reception time of the 1 time-stamp packet lies with the left and right uncertainty interval limits and is given by:

$$\underline{O}_x(1) = \max\left(t_x(1) + O_x + C,\ 0\right) \tag{49}$$

$$\overline{O}_x(1) = \min\left(t_x(1) + O_x + D,\ \infty\right). \tag{50}$$

considering the receiver sensor node starting time r_x can be anywhere between [0, ∞]. It cannot take a negative value since the clock runs forward rather than backward.

On receiving the 2 time-stamp packet

$$\underline{O}_x(2) = \max\left(t_x(2) + O_x + C,\ \underline{O}_x(1) + r_x(2) - r_x(1)\right) \tag{51}$$

$$\overline{O}_x(2) = \min\left(t_x(2) + O_x + D,\ \overline{O}_x(1) + r_x(2) - r_x(1)\right). \tag{52}$$

Substituting $\underline{O}_x(1)$ and $\overline{O}_x(1)$ from Eqs. (49) and (50), and $r_x(2)$ and $r_x(1)$ using Eq. (2) it can be rearranged as

$$\underline{O}_x(2) = O_x + \max\left(t_x(2) + C,\ t_x(2) + C + R(2) - R(1)\right) \tag{53}$$

$$\overline{O}_x(2) = O_x + \min\left(t_x(2) + D,\ t_x(2) + D + R(2) - R(1)\right) \tag{54}$$

rewriting it again as

$$\underline{O}_x(2) = O_x + t_x(2) + C + R(2) - \min\left(R(2),\ R(1)\right) \tag{55}$$

$$\overline{O}_x(2) = O_x + t_x(2) + D + R(2) - \max\left(R(2),\ R(1)\right). \tag{56}$$

Then, on receiving n time-stamped message packets, the reception time lies with the updated left and right limits and is given by:

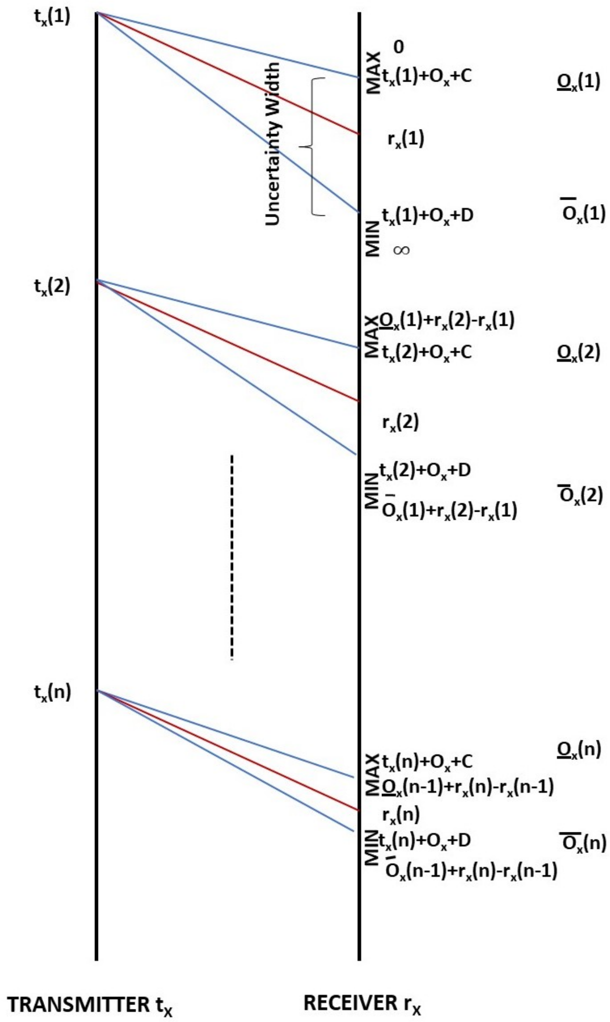

***Fig. 7**: Interval-based time coordination protocol analysis.*

$$\underline{O}_x(n) = O_x + t_x(n) + C + R(n) - \min\left(R(n),\ R(n-1),\ \ldots,\ R(1)\right) \tag{57}$$

$$\overline{O}_x(n) = O_x + t_x(n) + D + R(n) - \max\left(R(n),\ R(n-1),\ \ldots,\ R(1)\right). \tag{58}$$

After n transmissions, the ambiguity interval $\widetilde{O}(n)$ can be updated as:

$$\begin{aligned}\widetilde{O}_x(n) &= \overline{O}_x(n) - \underline{O}_x(n) \\ &= (D - C) - \Big(\max\left(R(n),\ R(n-1),\ \ldots,\ R(1)\right) \\ &\qquad - \min\left(R(n),\ R(n-1),\ \ldots,\ R(1)\right)\Big).\end{aligned} \tag{59, 60}$$

REFERENCES

[1] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[2] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, "A critical analysis of research potential, challenges, and future directives in industrial wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 39–95, 2018.

[3] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, May 2005.

[4] Y. Hlaing and N. A. M. Maung, "Hybrid time synchronization for ZigBee networks: An empirical approach," in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020, pp. 376–379.

[5] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, Jul. 2004.

[6] P. Ranganathan and K. Nygard, "Time synchronization in wireless sensor networks: A survey," *International Journal of UbiComp*, vol. 1, no. 2, pp. 92–102, Apr. 2010.

[7] M. A. Sarvghadi and T.-C. Wan, "Message passing based time synchronization in wireless sensor networks: A survey," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, May 2016, Art no. 1280904.

[8] N. Dalwadi and M. Padole, "An insight into time synchronization algorithms in IoT," in *Data, Engineering and Applications*, R. K. Shukla, J. Agrawal, S. Sharma, and G. S. Tomer, Eds. Singapore: Springer, 2019, pp. 285–296.

[9] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[10] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the first international conference on embedded networked sensor systems (SenSys '03)*, 2003, pp. 138–149.

[11] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, 2004, pp. 39–49.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, Dec. 2002.

[13] J. Zhang, S. Lin, and D. Liu, "Cluster-based time synchronization protocol for wireless sensor networks," in *Algorithms and Architectures for Parallel Processing*, X.-H. Sun *et al.*, Eds. Cham, Switzerland: Springer, 2014, pp. 700–711.

[14] Z. Wang, P. Zeng, M. Zhou, D. Li, and J. Wang, "Cluster-based maximum consensus time synchronization for industrial wireless sensor networks," *Sensors*, vol. 17, no. 1, Jan. 2017, Art. no. 141.

[15] R. Solis, V. S. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 2734–2739.

[16] I. Shames and A. N. Bishop, "Distributed relative clock synchronization for wireless sensor networks," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 265–11 270, Jan. 2011.

[17] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, Sep. 2008.

[18] K. Römer, "Time synchronization in ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '01)*, 2001, pp. 173–182.

[19] P. Blum, L. Meier, and L. Thiele, "Improved interval-based clock synchronization in sensor networks," in *Proceedings of the third international symposium on Information processing in sensor networks (IPSN'04)*, 2004, pp. 349–358.

[20] L. Meier, P. Blum, and L. Thiele, "Internal synchronization of drift-constraint clocks in ad-hoc sensor networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '04)*, 2004, pp. 90–97.

[21] K. L. V. S. P. Sakuru and N. B. Rao, "An iterative node-pair time synchronization (INTS) for wireless sensor networks," in *Electronic Systems and Intelligent Computing*, P. K. Mallick, P. Meher, A. Majumder, and S. K. Das, Eds. Singapore: Springer, 2020, pp. 495–505.

[22] T. Hao, R. Zhou, G. Xing, M. W. Mutka, and J. Chen, "WizSync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1379–1392, Jun. 2014.

[23] F. Asgarian and K. Najafi, "Time synchronization in a network of bluetooth low energy beacons," in *Proceedings of the SIGCOMM Posters and Demos*, 2017, pp. 119–120.

[24] A. Rowe, V. Gupta, and R. R. Rajkumar, "Low-power clock synchronization using electromagnetic energy radiating from AC power lines," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, 2009, pp. 211–224.

[25] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*, 3rd ed. Delhi, India: Pearson Education, 2003.

[26] S. N. Gelyan, A. N. Eghbali, L. Roustapoor, S. A. Y. F. Abadi, and M. Dehghan, "SLTP: Scalable lightweight time synchronization protocol for wireless sensor network," in *Mobile Ad-Hoc and Sensor Networks. MSN 2007*, H. Zhang, S. Olariu, J. Cao, and D. B. Johnson, Eds. Berlin, Germany: Springer, 2007, pp. 536–547.

K. L. V. Sai Prakash Sakuru received his B.E. in Electronics and Communication Engineering from Andhra University, India and M. Tech from Regional Engineering College, Warangal, India. He is currently working as Associate Professor at Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, India. His current research interests include Wireless sensor networks, Internet of Things, Resource allocation, and energy analysis.

N. Bheema Rao received his B.E. in Electronics and Communication Engineering from Andhra University, India, M. Tech from Andhra University, India, and Ph.D. from Indian Institute of Technology, Bombay, India. He is currently working as Professor at Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, India. He has published over 60 articles in different journals & conferences. He guided 5 Ph.D. scholars and is presently guiding 4 scholars in the area on chip passive components for RF Applications, Device Modelling, Interconnect Modelling and Sensor networks.