

**Impact of Linear Regression on Time
Synchronization Accuracy and Energy Consumption
for Wireless Sensor Networks**

Copyright © November, 2008, IEEE. Reprinted from Military Communications Conference, 2008 (IEEE MILCOM '08).

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Argon ST's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Impact of Linear Regression on Time Synchronization Accuracy and Energy Consumption for Wireless Sensor Networks

Liangping Ma, Hua Zhu, Gayathri Nallamothu, Bo Ryu, and Zhensheng Zhang
Network Systems, Argon ST, Inc.
San Diego, CA 92121

Email: {lma, hua.zhu, gayathri.nallamothu, bo.ryu, zhensheng.zhang}@argonst.com

Abstract—Linear regression is used in a number of time synchronization protocols to establish a linear relationship between clocks at different sensor nodes to achieve energy-efficient time synchronization. These protocols are predictive in the sense that a node predicts a target clock based on collected time stamp data. However, the use of linear regression for sensor network time synchronization has not been thoroughly studied in the literature. This paper attempts to close the gap by analyzing the impacts of two parameters on the precision of linear regression based time synchronization protocols: (1) the frequency at which the time stamp data are collected; (2) the window size, i.e., the number of time stamps used for linear regression. Through theoretical analysis, experiments and simulations, we show a counter-intuitive result: given the prediction interval, if the clock relationship varies slowly over time, more frequent synchronization results in worse synchronization precision. This result suggests that a linear regression based time synchronization protocol can achieve both high precision and good energy efficiency when operating at a low synchronization frequency. We also show that increasing the window size improves the synchronization performance but the synchronization uncertainty is bounded away from zero.

I. INTRODUCTION

A common misunderstanding about linear regression based time synchronization protocols is that, the higher the time synchronization frequency, the better the synchronization precision. The time synchronization frequency dictates how often time synchronization messages are exchanged and time stamp data are collected. This misunderstanding seems to agree with the intuition that the more frequently the data are sampled, the better the true regularity or pattern is extracted. However, this is not true. In this paper, we show, through theoretical analysis, experiments and simulations, that given a fixed window size and a fixed prediction interval, the opposite

of the intuition is true. That is, a higher synchronization frequency results in worse synchronization precision. In addition, we analyze the impact of the window size on the synchronization precision. The background of our work is as follows.

Time synchronization is important to wireless sensor networks. For one thing, a common time reference is necessary for such applications as target tracking. For another, to conserve energy, duty-cycling (i.e., the radios are not always on, but instead on and off in an alternating fashion) is widely used, and time synchronization enables successful rendezvous of the transmitter and the receiver. The use of time synchronization protocols is a practical approach to achieving energy efficient and robust time synchronization in wireless sensor networks. In contrast, the two alternative approaches, namely, GPS and atomic clocks, have major drawbacks. The GPS signal is not always available to a wireless sensor network due to shadowing and jamming. Atomic clocks, although extremely stable and accurate, consume too much energy and are not energy efficient.

There are two types of time synchronization protocols: instant synchronization and predictive synchronization. In instant time synchronization protocols, a node corrects its own clock reading by adopting its neighbor's. If the two clocks run at the same rate, a single synchronization point will suffice. However, clocks run at different rates, and the resulting difference in clock readings accumulates over time [2]. Thus to maintain high precision, many synchronization points are needed, meaning a high energy expenditure. Instant time synchronization protocols include RBS [3], TPSN [4], and time diffusion [5]. In predictive time synchronization protocols, each node tries to establish a relationship between its own clock and a target clock. A simple model for this relationship is a linear model, whose optimal solution is given by

linear regression [1]. Predictive synchronization greatly reduces the number of synchronization points and hence the energy used for time synchronization, even for clocks exhibiting dramatic differences. Predictive synchronization protocols include FTSP [6] and RATS [7].

The remainder of this paper is organized as follows. Section II formulates the time synchronization problem and fits it into the linear regression theory. Section III gives the theoretical analysis. Section IV shows sensor network testbed experimental results, and Matlab and QualNet simulation results. Section V concludes this paper.

II. TIME SYNCHRONIZATION AND LINEAR REGRESSION

A. The Time Synchronization Problem

Consider two wireless sensor nodes 1 and 2. Denote their clock readings as $x(t)$ and $y(t)$, respectively, where t is the ideal time, and $x(t)$ and $y(t)$ are non-decreasing functions of t . The goal of time synchronization is for one node to establish a clock relationship that allows it to translate the other node's clock reading into its own. The clock translation provides an important means for the two nodes to agree upon a future time in the absence of a common time reference such as the GPS timing signals or extremely stable clock sources such as atomic clocks.

To formulate the problem, we consider how time synchronization is done between two sensor nodes in practice. Due to symmetry, we consider how node 2 synchronizes with node 1 only. The procedure is as follows. Node 1 generates a time stamp $x(t_1)$ at time t_1 and puts it in a message called a *synch message*, which is then sent to node 2. Of course, node 1 does not know t_1 since it has no access to an ideal clock. Upon receiving the synch message at time $t'_1 > t_1$, node 2 generates a new time stamp $y(t'_1)$. If node 2 knew the exact value of $t'_1 - t_1$ and if node 2's clock progressed at the same rate as an ideal clock, node 2 would adjust its time stamp $y(t'_1)$ to $y(t_1) = y(t'_1) - (t'_1 - t_1)$, which corresponds to $x(t_1)$. However, the exact adjustment will never be known for sure in practice. The actual adjusted time stamp can be modeled as a random variable

$$Y(t_1) = y(t'_1) - (t'_1 - t_1) + E(t_1, t'_1), \quad (1)$$

where $E(t_1, t'_1)$ represents the random error in adjusting the time stamp. This way, node 2 collects a time stamp pair $(x(t_1), Y(t_1))$.

Similarly, node 1 generates time stamps at time t_2, t_3, \dots and sends the respective synch messages to node 2, which collects time stamp

pairs $(x(t_2), Y(t_2)), (x(t_3), Y(t_3)), \dots$. In general, we have

$$Y(t) = y(t') - (t' - t) + E(t, t'), \quad (2)$$

or equivalently

$$Y(t) = y(t) + (y(t') - y(t)) - (t' - t) + E(t, t'). \quad (3)$$

When the clock of node 2 is stable and runs ρ times as fast as an ideal clock, which is approximately true for a good quality quartz clock in a stable environment, we have $y(t') - y(t) = \rho(t' - t)$. The time difference $t' - t$ consists of the propagation delay, transmission time, which can be thought of as fixed in a stationary wireless sensor network with fixed packet size, and the average packet processing time. Therefore, $(\rho - 1)(t' - t)$ can be thought of as a constant, and (3) can be rewritten as

$$Y(t) = y(t) + c + E(t, t'), \quad (4)$$

where

$$c = (\rho - 1)(t' - t). \quad (5)$$

Let the relationship between $x(t)$ and $y(t)$ be a smooth function $y(t) = f(x(t))$. Then, by the Taylor expansion, there exists a point $x(t_0)$ with a neighborhood $N(x(t_0)) \ni x(t)$ such that

$$y(t) = f(x(t_0)) + f'(x(t_0))(x(t) - x(t_0)) + o(x(t) - x(t_0)). \quad (6)$$

Substituting (6) into (4) yields

$$Y(t) = \alpha(t_0) + \beta(t_0)x(t) + E(t, t'), \quad (7)$$

where

$$\alpha(t_0) = f(x(t_0)) - f'(x(t_0))x(t_0) + o(x(t) - x(t_0)) + c, \quad (8)$$

$$\beta(t_0) = f'(x(t_0)), \quad (9)$$

and $E(t, t')$, as explained earlier, represents the uncertainty in the time stamp adjustment, which we model as a white Gaussian stochastic process with mean zero and variance σ^2 , i.e., $E(t, t') \sim \mathcal{N}(0, \sigma^2)$. For ease of exposition, we rewrite (7) as

$$Y = \alpha + \beta x + E, \quad (10)$$

where $E \sim \mathcal{N}(0, \sigma^2)$, and denote

$$x_i = x(t_i), \quad (11)$$

$$Y_i = Y(t_i), \quad (12)$$

$$E_i = E(t_i, t'_i), \quad (13)$$

for $i = 1, 2, \dots, n$.

The time synchronization problem is stated as follows. Given an *observation set* $\{(x_i, Y_i) | i = 1, 2, \dots, n\}$, infer

α and β in the linear model (10) to establish an estimator of Y :

$$\hat{Y} = \hat{\alpha} + \hat{\beta}x, \quad (14)$$

which can serve as a clock translator.

B. The Optimal Solution

The optimal estimators $\hat{\alpha}$ and $\hat{\beta}$ can be obtained by the standard linear regression theory,

$$\hat{\beta} = \frac{n \sum x_i Y_i - \sum x_i \sum Y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (15)$$

$$\hat{\alpha} = \frac{\sum Y_i}{n} - \hat{\beta} \frac{\sum x_i}{n} \quad (16)$$

which maximize the likelihood of the observation set [8]. Here all summations run from 1 to n . To translate a future time of node 1, x^* , into node 2's own time, node 2 uses (14) to get

$$\hat{Y}^* = \hat{\alpha} + \hat{\beta}x^*, \quad (17)$$

and the actual Y corresponding to x^* , i.e., Y^* , will fall within the following interval

$$(\hat{Y}^* - w, \hat{Y}^* + w) \quad (18)$$

with probability $100(1 - \gamma)\%$, where

$$w(x^*) = t_{\gamma/2, n-2} \sqrt{\frac{\sum (Y_i - \hat{Y}_i)^2}{n-2}} \times \sqrt{1 + \frac{1}{n} + \frac{(x^* - \sum \frac{x_i}{n})^2}{\sum (x_i - \sum \frac{x_i}{n})^2}} \quad (19)$$

and

$$\hat{Y}_i = \hat{\alpha} + \hat{\beta}x_i. \quad (20)$$

The number $t_{\gamma/2, n-2}$ is half of the length of the interval centered at the mean of a Student distribution of degree of freedom $n-2$ over which the probability adds to $1-\gamma$. As an example, for $\gamma = 0.05$ (i.e., a 95% confidence interval) and $n = 4$, $t_{\gamma/2, n-2} = 4.303$.

III. ANALYSIS

In this section, we evaluate the performance of linear regression from three aspects: (1) At what time scale a linear clock model is valid? (2) Given the linear model, what determines the performance of the clock translation/prediction and in what manner? (3) What practical issues affect the prediction performance?

We consider a particular time synchronization scheme used in a few time synchronization algorithms [6] [7]:

$$x_i = x_1 + (i - 1)d, \quad i = 1, 2, \dots, n, \quad (21)$$

where n is called the *window size* and d is called the *synch interval*. That is, node 1 sends a synch message to node 2 every d seconds. To reduce the computational complexity, x_1 is subtracted from all x_i , which effectively sets

$$x_i = (i - 1)d, \quad i = 1, 2, \dots, n, \quad (22)$$

which are the input to linear regression.

We define the *prediction interval* p as the time difference between the time when the prediction is made (i.e., when the latest time stamp pair (x_n, Y_n) is collected), and the time when the predicted event occurs. In other words, $p = x^* - x_n$.

A. Validity of the Linear Model

As discussed in Section II-A, if the relationship between $x(t)$ and $y(t)$ is characterized by a smooth function $y(t) = f(x(t))$, then for any point $x(t_0)$, we can always find a neighborhood in which a linear relationship is arbitrarily accurate. The radius of the neighborhood may depend on $x(t_0)$. On the other hand, if we do not have the freedom to choose the size of the neighborhood, is a linear model still valid? The answer depends on the time scale of the clock dynamics $f(x(t))$ and the prediction interval p . If the time scale is larger than p , a linear model is accurate, as shown in Fig. 1(a). Otherwise, a linear model may cause excessive errors, as shown in Fig. 1(b).

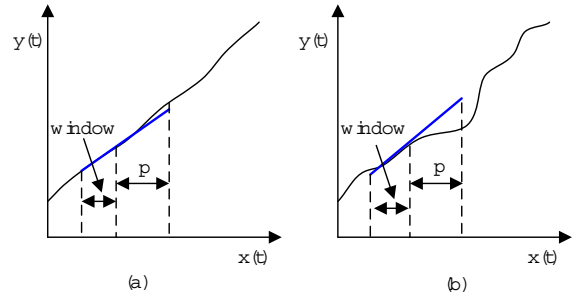


Fig. 1. Accuracy of a linear model depends on the time scale of the clock dynamics and the prediction interval p . The window includes all the time stamps used to determine the linear model.

B. Theoretical Results on the Performance of the Prediction

We define:

- the *prediction uncertainty* as $w(x^*)$ in (19),
- and the *root mean square (RMS) prediction uncertainty* as $u(x^*) = \sqrt{E[w^2(x^*)]}$, which measures the

performance of the prediction when the distribution of E_i is considered.

The theoretical results are as follows.

Theorem 3.1: Under the linear model (10) and the time synchronization scheme (22), if the observation set $\{(x_i, Y_i) | i = 1, 2, \dots, n\}$ is fixed, the prediction uncertainty $w(x^*)$ depends only on the ratio of the prediction interval p to the synch interval d , i.e., p/d .

Proof: By (21), we have

$$\sum_{i=1}^n \frac{x_i}{n} = x_1 + \frac{n-1}{2}d. \quad (23)$$

Therefore

$$\frac{(x^* - \sum_{i=1}^n \frac{x_i}{n})^2}{\sum_{i=1}^n (x_i - \sum_{j=1}^n \frac{x_j}{n})^2} = \frac{3(n-1 + 2p/d)^2}{(n-1)n(n+1)}. \quad (24)$$

Since the observation set is given, its size n is fixed and thus (24) depends only on p/d . Also, $\hat{\beta}$ in (15) and $\hat{\alpha}$ in (16) are fixed, so is \hat{Y}_i in (20). Thus, $\sum (Y_i - \hat{Y}_i)^2$ is fixed. It follows from (19) that $w(x^*)$ depends only on p/d . \square

Lemma 1: Under the linear model (10) and the time synchronization scheme (22), if n is fixed, $E[\sum_{i=1}^n (Y_i - \hat{Y}_i)^2]$ is invariant of the synch interval d .

Proof: Since $E(t, t')$ in (7) is a white Gaussian stochastic process, $E_i, i = 1, \dots, n$ are independent.

$$\begin{aligned} & E\left[\sum_{i=1}^n (Y_i - \hat{Y}_i)^2\right] \\ &= \sum_{i=1}^n E\left[\left((\alpha + \beta x_i + E_i) - (\hat{\alpha} + \hat{\beta} x_i)\right)^2\right] \quad (25) \end{aligned}$$

$$= \sum_{i=1}^n E\left[\left((\alpha - \hat{\alpha}) + (\beta - \hat{\beta})x_i + E_i\right)^2\right] \quad (26)$$

$$= \sum_{i=1}^n [\text{var}(\hat{\alpha}) + \text{var}(\hat{\beta})x_i^2 + \text{var}(E_i)] \quad (27)$$

$$= n\text{var}(\hat{\alpha}) + \text{var}(\hat{\beta}) \sum_{i=1}^n x_i^2 + n\text{var}(E_i) \quad (28)$$

$$= \frac{\sigma^2 \sum_{i=1}^n x_i^2}{\sum_{i=1}^n (x_i - \sum_{j=1}^n x_j/n)^2} + \frac{\sigma^2 \sum_{i=1}^n x_i^2}{\sum_{i=1}^n (x_i - \sum_{j=1}^n x_j/n)^2} + n\sigma^2 \quad (29)$$

$$= \frac{2\sigma^2 \sum_{i=1}^n x_i^2}{\sum_{i=1}^n (x_i - \sum_{j=1}^n x_j/n)^2} + n\sigma^2, \quad (30)$$

where the fact that $\hat{\alpha}, \hat{\beta}$ and E_i are uncorrelated is used in (27). Substituting (22) into (30), we see that d is cancelled and (30) depends on n , which is fixed. \square

Theorem 3.2: Under the linear model (10) and the time synchronization scheme (22): (i) the RMS prediction uncertainty $u(x^*) > t_{\gamma/2, n-2} \sqrt{1 + 1/n} \sigma, \forall p, d$; (ii) if n is fixed, $u(x^*)$ increases with p/d .

Proof: (i) From (19), we have

$$\begin{aligned} E[w^2(x^*)] &= t_{\gamma/2, n-2}^2 \frac{E[\sum (Y_i - \hat{Y}_i)^2]}{n-2} \left(1 + \frac{1}{n} \right. \\ &\quad \left. + \frac{(x^* - \sum \frac{x_i}{n})^2}{\sum (x_i - \sum \frac{x_j}{n})^2}\right). \quad (31) \end{aligned}$$

By (30) and (22), we have

$$\begin{aligned} \frac{E[\sum_{i=1}^n (Y_i - \hat{Y}_i)^2]}{n-2} &= \frac{2(2n-1)\sigma^2}{(n-2)(n+1)} + \frac{n\sigma^2}{n-2} \quad (32) \\ &> \sigma^2, \quad (33) \end{aligned}$$

which together with (31) and the definition of $u(x^*)$ gives $u(x^*) > t_{\gamma/2, n-2} \sqrt{1 + 1/n} \sigma, \forall p, d$.

(ii) Since n is fixed, by Lemma 1, $E[\sum (Y_i - \hat{Y}_i)^2]$ is fixed. By (24), the term in the parentheses in (31) increases with p/d . So does $E[w^2(x^*)]$, and $u(x^*)$. \square

Note: Part (ii) of Theorem 3.2 says that we can reduce the prediction uncertainty by increasing the synch interval d , but Part (i) says that the prediction uncertainty will always be greater than σ times a constant.

Theorem 3.3: Under the linear model (10) and the time synchronization scheme (22): (i) if n and p are fixed, the RMS prediction uncertainty $u(x^*)$ decreases as d increases; (ii) if p/d is fixed, $u(x^*)$ decreases as n increases, asymptotically $u(x^*) = \Theta((\sigma^2 + 4(1 + \sigma^2)/n)^{1/2})$, and in the limit $u(x^*) \rightarrow t_{\gamma/2, \infty} \sigma$ as $n \rightarrow \infty$.

Proof: (i) This follows from Theorem 3.2 by considering d only in the ratio p/d .

(ii) By (32), $E[\sum_{i=1}^n (Y_i - \hat{Y}_i)^2]/(n-2)$ decreases as n increases. By (24), $1 + 1/n + (x^* - \sum \frac{x_i}{n})^2 / \sum (x_i - \sum \frac{x_j}{n})^2$ decreases as n increases. Also, $t_{\gamma/2, n-2}$ decreases as n increases [8]. Therefore, by (31), $E[w^2(x^*)]$ and hence $u(x^*)$ decrease as n increases.

As $n \rightarrow \infty$, the Student distribution approaches a standard Gaussian distribution, and thus $t_{\gamma/2, n-2}$ approaches a constant that depends only on γ , which is fixed for a given confidence. The limit result follows from (24) and (32). \square

Note: The asymptotic result says that the prediction uncertainty $u(x^*)$ decays very slowly with the window size n when n is large. The limit result $u(x^*) \rightarrow t_{\gamma/2, \infty} \sigma$ says

that it is impossible to reduce the prediction uncertainty to zero by increasing the window size infinitely.

Part (i) of Theorem 3.3 is illustrated in Fig. 2(a). Consider two nodes 1 and 2. Node 2 translates a future clock reading of node 1 into its own clock reading. This translation is essentially prediction. We compare two scenarios a and b with prediction intervals $p_a = p_b = 2T$ and time synch intervals $d_a = 2T$ and $d_b = T$. The red dashed lines are for scenario b : the leftmost one represents the prediction uncertainties for predictions made at time 0, and the second leftmost one for predictions made at time T , and so on. The blue solid lines are for scenario a : the leftmost one represents the prediction uncertainties for predictions made at time 0, and the second leftmost one for predictions made at time $2T$, and so on. Looking at time $2T$ on the horizontal axis, we see that the prediction uncertainty w_a of scenario a is less than w_b , that of scenario b . That is, when the synch messages are sent less frequently, less prediction uncertainty and hence better synchronization precision is achieved.

Theorem 3.1 is illustrated in Fig. 2(b). Now we change $p_b = T$, and keep $p_a = 2T$. That is, the ratio p/d is equal to 1 for both scenarios. The same prediction uncertainty w_a is achieved in both scenarios. That is, the prediction uncertainty does not depend on how frequently the synch messages are sent if the prediction interval is equal to the synch interval.

The results illustrated above are counter-intuitive, but they are the consequences of the clock model (10) and linear regression. In the Rate Adaptive Time Synchronization (RATS) algorithm [7], the synch interval is adjusted based on the computed prediction uncertainty. When the prediction uncertainty is higher than a threshold η_{\max} , the synch interval is halved to increase the synchronization frequency, and when the prediction uncertainty is lower than a threshold $\eta_{\min} < \eta_{\max}$, the synch interval is doubled to decrease the synchronization frequency. The rationale behind RATS is the intuition that more frequent synchronization results in higher synchronization precision, which is not true at least in the case where the clocks follow a linear model (10). As a result, when the prediction uncertainty is higher than η_{\max} , the rate adaptation will keep halving the synch interval, as we observed in the Mica2 experiments, until reaching a minimal allowable value. This not only worsens the synchronization precision, but also wastes energy.

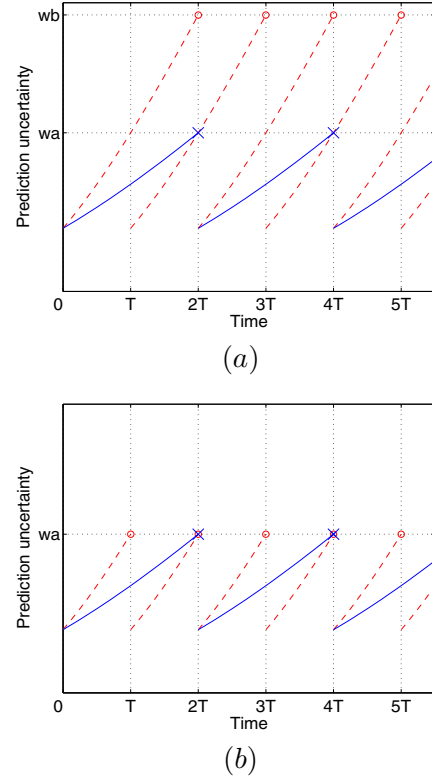


Fig. 2. (a) If the prediction interval is fixed, more frequent synchronization results in more prediction uncertainty or worse synch performance: the red dashed lines are for $d = T$, the blue solid lines for $d = 2T$, and the predictions are made at integer multiple of T . (b) If the prediction interval is equal to the synch interval, i.e., $p = d$, the same synch performance is achieved regardless of the value of d : The red dashed lines are for $d = T$, the blue solid lines for $d = 2T$, and predictions are made at integer multiples of T and $2T$ respectively.

C. Practical Issues on the Prediction Performance

For typical quartz clocks, the slope β is very close to 1 and takes values like 1.00000737. The difference from 1, such as 0.00000737 here, although very small, is important to capturing the difference in the frequency offset [2] of two clocks. If it were ignored, we would be assuming that two clocks are running at the same rate, which will lead to significant accumulative errors in time synchronization.

To obtain a meaningful value for β , linear regression should not be implemented using integer computation. However, the single floating point data type, which is the only floating point data type available at a typical microsensor node, is insufficient to represent numbers of many significant digits such as 1.00000737, because it provides a precision of only 7 decimal significant digits (or 23 binary significant digits) [9]. Therefore, for processors that do not support double floating point

computation, significant truncation errors will occur.

A truncation error also occurs when the time stamp is converted from one data type to another. In our Mica2 [10] testbed, the time stamp is represented by a 32 bit unsigned integer. If we convert it to the single floating point, the precision will be reduced to 23 binary significant digits, resulting in a significant truncation error.

IV. SIMULATION AND EXPERIMENTAL RESULTS

We implement linear regression for time synchronization in a Mica2 microsensor network testbed, where the Mica2 [10] nodes run Sensor Operating System (SOS) [11], use the low power 916MHz CC1000 radio, the 8-bit 7.37-MHz Atmega 128L microprocessor as the CPU, have 128KB program memory, and support a maximum raw data rate of 38.4kbps.

Due to the lack of support for double floating point computation in the Atmega processor, we collect time stamp data from a sensor network testbed, and then resort to Matlab and QualNet [12] network simulator for high precision simulations.

A. Experimental Results

The sensor network testbed for the time stamp collection is shown in Fig. 3:

- The testbench consists of a laptop, a 4-port USB serial adaptor, and two Mica2 sensor nodes. The 4-port USB Serial adapter acts as a gateway between the laptop (using a USB port) and the 2 Mica2 sensor nodes (each using a serial port).
- The two Mica2 nodes exchange time synch messages.
- Each node can print information, including the neighbor list, time stamp pairs and linear regression results, in a Cygwin window on the laptop.
- A time synch message is sent every 30 seconds.

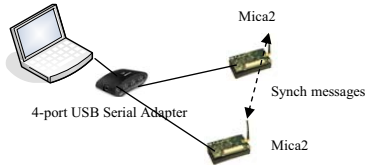


Fig. 3. Experiment set up.

The above experiment provides time stamp data as 32 bit unsigned integers. As mentioned in Section III-C, the lack of support for double floating point computation leads to significant truncation errors. To address this

problem, we convert the time stamp data into double floating point and then do linear regression in Matlab.

Figure 4 shows that if the prediction interval p is fixed at 600s, the prediction uncertainty decreases as the time synchronization interval d increases, and the experimental result (solid blue line) agrees with the QualNet simulation result (red dashed line). Next, we let the prediction interval equal to the time synchronization interval, i.e., $p = d$, and the results are shown in Fig. 5(a), where the prediction uncertainty fluctuates around a constant. The fluctuation becomes more severe as the time synchronization interval increases because the larger the interval the fewer time stamps are available. However, as explained in Section III-A, depending on the time scale of the clock dynamics, this invariance phenomenon may not hold for large d . Finally, Fig. 5(b) suggests that the prediction uncertainty approaches to a constant above zero as the window size increases, which agrees with Theorem 3.3.

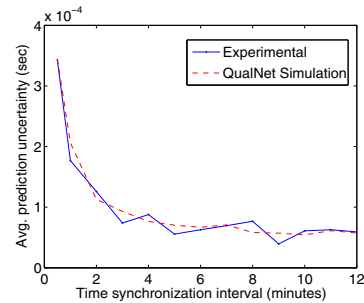


Fig. 4. The impact of the time synchronization interval on the prediction uncertainty for prediction interval $p = 600$ seconds.

B. QualNet Simulation Results

We devised an innovative method [13] for realistic clock modeling in any discrete-event simulator, including QualNet [12]. We generate a realization of a random clock model for each node, which maps the ideal time (or simulation time) to each node's local time. These mappings are reversible. The time of each event, which is in a node's local clock, is translated into the simulation time, allowing QualNet to simulate the events from different nodes in the right order. Refer to [13] for more details. Among many options, a simplified clock model defined in [2] is used in the following simulations. The values of the parameters that match the clock characterization of Mica2 nodes are identified. The QualNet simulation results well match the corresponding experimental results, as shown in Fig. 4 and Fig. 5(a).

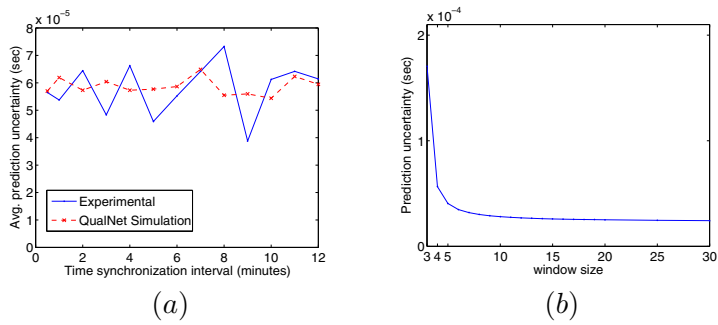


Fig. 5. (a) The impact of the time synchronization interval on the prediction uncertainty for $p = d$, where d varies from 0.5 minutes to 12 minutes; (b) The impact of the window size on the prediction uncertainty, with fixed synchronization interval 30 seconds (= prediction interval).

V. CONCLUSION

In this paper, we investigate the performance of linear regression in predictive time synchronization protocols for wireless sensor networks. Basing on theoretical analysis, experiments and simulations, we show that, when the clock relationship changes slowly over time, for a fixed prediction interval, the more frequently the synchronization messages are exchanged, the worse the synchronization precision. Therefore, in practice, there is no need to synchronize frequently, and a low synchronization frequency offers both high synchronization precision and good energy efficiency. In addition, as the window size increases, the time synchronization precision improves but the synchronization uncertainty is bounded away from zero. However, when the clock relationship changes quickly over time, the time synchronization frequency should be high enough to track the changing clock relationship. We leave this case for future research.

ACKNOWLEDGEMENT

We would like to thank Professor Mani Srivastava of UCLA and his research group, including Ilias Tsigkogiannis, Saurabh Ganeriwal, and Simon Han for their invaluable support.

REFERENCES

- [1] L. Ma, H. Zhu, G. Nallamothu, B. Ryu, and H. R. Howard, "Understanding linear regression for wireless sensor network time synchronization," in *International Conference on Wireless Networks (ICWN'07)*, (Las Vegas, Nevada), 2007.
- [2] D. W. Allan, "Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators," *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, pp. 647–654, Nov. 1987.

- [3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th symposium on Operating systems Design and Implementation (OSDI)*, 2002.
- [4] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *The 1st ACM Conference on Embedded Networked Sensor Systems*, (Los Angeles, California), Nov. 2003.
- [5] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *INFOCOM*, 2004.
- [6] M. Marti, B. Kusy, G. Simon, and . Ldeczi, "The flooding time synchronization protocol," in *The 2nd ACM Conference on Embedded Networked Sensor Systems*, (Baltimore, Maryland), Nov. 2004.
- [7] S. Ganeriwal, D. Ganesan, H. Sim, V. Tsitsis, and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *The 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, (San Diego, California), Nov. 2005.
- [8] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*. McGraw-Hill Inc., 1974.
- [9] S. Brown and Z. Varanasic, *Fundamentals of Digital Logic with VHDL design*. McGraw-Hill, 2000.
- [10] Crossbow Technology, Inc., "Product features: Mica2 433, 868, 916 mhz." [Online]. Available: <http://www.xbow.com/Products/productdetails.aspx?sid=174>.
- [11] UCLA, "Sensor operating system." [Online]. Available: <http://nesl.ee.ucla.edu/projects/sos>.
- [12] Scalable Network Technologies, Inc., "Qualnet network simulator." [Online]. Available: <http://www.scalable-networks.com>.
- [13] H. Zhu, L. Ma, and B. Ryu, "Clock modeling in discrete-event simulation," 2007. US Provisional Patent No.60/929,121.