

# On the Multi-Hop Performance of Synchronization Mechanisms in High Propagation Delay Networks

Pai-Han Huang, *Electrical Engineering-Systems, University of Southern California*

Maulik Desai, *Electrical Engineering, Columbia University*

Xiaofan Qiu, *Texas Instruments Inc.*

Bhaskar Krishnamachari, *Electrical Engineering-Systems, University of Southern California*

**Abstract**—We analyze the single and multi-hop performance of time synchronization mechanisms for challenging environments characterized by high propagation delays, low duty-cycle operation, and imprecise clocks, such as underwater acoustic sensor networks. We find that receiver-receiver based schemes are unsuitable for such environments, and therefore focus primarily on sender-receiver schemes. According to our analysis, a one-way dissemination approach provides good clock skew estimation but poor offset estimation while a two-way exchange approach provides accurate offset estimation but imprecise clock skew estimation. In average, using one-way scheme can result in significant cumulative propagation error over multiple hops, and using two-way can lead to high variance of propagation error. We develop and analyze a hybrid one-way dissemination/two-way exchange technique, and verify the performance of our hybrid scheme through trace-based experiments. The results suggest that this hybrid approach can provide bounded average error propagation in multi-hop settings and significantly lower variance of propagation error.

**Index Terms**—Wireless communications, Formal models, Data communications, Protocol verification, Sensor networks, Algorithm/protocol design and analysis.



## 1 INTRODUCTION

Recently there has been an emergence of interest in developing sensor networks that operate in even more challenging environments, such as acoustic underwater networks, that are characterized by high propagation delays, high relative clock skews, and longer inter-sync periods (due to even more severe energy constraints). In this work, we analyze various existing approaches for time synchronization in RF-based sensor networks and find that they can yield significant error under these stricter conditions. Our analysis suggests that a hybrid mechanism does much better in terms of precision and variance.

The packets used for synchronization purpose are called synchronization packets. Depending on the synchronization packets exchange strategies, existing works can be categorized into two major types. One is the sender-receiver scheme, which involves direct communication between two devices in order to synchronize one to the other. The other one is the receiver-receiver scheme, which will synchronize clocks between nodes receiving the same reference broadcast. The existing works that involve sender-receiver synchronization can

be further categorized into two types. One can be called *one-way dissemination* and the other is *two-way exchange*. While the former needs a synchronized node to disseminate packets to unsynchronized nodes, the latter achieves synchronization by exchanging packets between synchronized and unsynchronized nodes. FTSP [2] and TPSN [3] are representative works which utilize one-way packet dissemination scheme and two-way packet exchange scheme respectively, and RBS [4] is a representative work of receiver-receiver scheme.

Clock offset and skew are two critical parameters while doing time synchronization. Clock offset refers to the difference of two clocks: the clocks within the reference node and the node that needs synchronization, at the moment of synchronization. Estimation of the offset thus helps to reset the clocks for re-synchronization. Clock skew refers to change in this offset over some period of time. Estimating the rate of this change can be useful in mitigating further clock drift between synchronization events, especially in a sensor network adopting low duty cycle. We undertake a comparative analysis of the synchronization error of various synchronization schemes in this paper. According to our analysis, in high propagation delay environments, RBS-like receiver-receiver schemes show poor performance. Among the sender-receiver schemes, the one-way packet dissemination does better on estimating clock skew, while the two-way exchange does better on estimating clock offset. However, in high propagation delay environments, the

The work described here is supported in part by NSF through grants CNS-0347621, CNS-0627028, CCF-0430061, CNS-0325875, NASA through an AIST grant, and the USC Viterbi School of Engineering's Summer Internship program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, NASA, or USC Viterbi School of Engineering.

one-way scheme shows high average synchronization error while the two-way scheme shows high variance.

We use our insights from the analysis to propose a hybrid one-way/two-way mechanism that is shown theoretically to perform more gracefully in multi-hop networks with high propagation delay, while incurring mild additional communication overhead. We validate our analysis through experimental trace-based simulations on a line network of radio-based sensor nodes where we use application-layer time-stamping to mimic conditions of high propagation delay. The results of these simulations confirm that the hybrid scheme not only provides bounded error propagation over multiple hops in average, but also produces low variance of propagation error.

The key contributions of this paper are threefold. First, we present comprehensive analysis of various existing synchronization schemes and address the pros and cons of each of them. We make use of our results and propose a hybrid scheme which provides precise and low variance synchronization. In addition, we provide both theoretical analysis and experimental trace-based simulations to verify the performance of our hybrid scheme.

This paper is organized as follows: we first present some existing efforts in Section 2, and analyze the precision for one-way, two-way, receiver-receiver, multiple two-way and the hybrid mechanism for a single sender-receiver pair in Section 3. We then investigate the multi-hop synchronization error propagation of one-way, two-way and hybrid in Section 4. Details of our time stamps collecting experiments are stated in Section 5. The three approaches are compared via trace based simulations in Section 6. We conclude with a summary and directions for future work in Section 7.

## 2 RELATED WORKS

Several issues make time synchronization challenging in high propagation delay environment, such as underwater acoustic networks. First, because the propagation distance of radio signal in water is limited, acoustic communication becomes a popular alternative [26]. However, the speed of underwater sound is five orders slower than radio signal, propagation latency becomes a non-negligible source of delay. Therefore, alleviating the impact of high propagation delay plays a major role in devising an efficient time synchronization protocol for such challenging environment. Because of low cost hardware adopted in sensor networks, clock generators perform differently from each other. Thus, eliminating the uncertainty is also an important and challenging issue. Some existing literatures [12] [27] have more detailed descriptions about underwater acoustic sensor networks.

Reference Broadcast Synchronization (RBS) [4] is an representative protocol of receiver-receiver scheme. In RBS, a reference beacon is broadcasted by a reference

node. Nodes which receive the broadcasted beacon will record its time of arrival and exchange this information with others. The average error of RBS is  $1.85 \pm 1.28 \mu\text{s}$  in 802.11 wireless Ethernet with kernel timestamping [4]. However, as shown in following contents, RBS is not suitable for high propagation delay environments due to the different distance between the reference node and other receivers.

Karp *et al.* [19] propose an variation of RBS, which is applicable to a network with a loop. The authors use resistive networks to derive a synchronization algorithm which provides global consistency and low variance. The trade-off between energy consumption and synchronization precision is also studied. Because we focus on tree-structured networks, this type of implementation is out of extent of our work.

Timing-sync Protocol for Sensor Networks (TPSN) [3] is one implementations of two-way exchange scheme. For a multi-hop network, TPSN first builds up a tree, assigns each node to a specific level, and does synchronization from level to level sequentially. The average error of TPSN is  $16.9 \mu\text{s}$  [3]. In addition, TPSN is the first protocol to use MAC layer time stamping, which is able to reduce medium access time efficiently. Lightweight Time Synchronization (LTS) [13] also adopts two-way exchange scheme and is a similar tree-style, sender-receiver implementation as TPSN.

Gridhar *et al.* [20] and Solis *et al.* [21] use two-way exchange scheme, and takes advantage of loops existing in networks to smoothen their synchronization estimation. The key idea of these papers is the sum of offset along a loop should be zero, and imposing these constraints to each loop can provide a smooth and accurate value.

Graham *et al.* [24] and Sichitiu *et al.* [14] propose synchronization protocols which use two-way exchange scheme. CTP [24] uses a similar equation as TPSN to estimate offset, and then further refines the estimation with linear regression. While Tiny-sync and Mini-sync [14] uses multiple time stamps collected from bi-directional synchronization packets to form constraints of clock offset and skew estimations, this algorithm has similar precision as CTP [24] when the number of time stamps is large.

The Flooding Time Synchronization Protocol (FTSP) [2] utilizes one-way dissemination scheme. Maroti *et al.* propose a sophisticated time stamping technique, which is able to reduce various kinds of jitter terms significantly [2]. The performance of FTSP is mainly verified by experiments, and the average error of FTSP is  $1.5 \mu\text{s}$  pair-wise and  $0.5 \mu\text{s}$  per hop [2] (Note: by “pair-wise” and “per hop”, we are referring to single-hop and multi-hop network setting, respectively. The “per-hop” error, is calculated by dividing the error at  $n$ -hop clock by  $n$ ).

RITS and RATS [23] [11] both utilize one-way scheme. One important conclusion is that if the non-determinism is minimal, even a simple implementation of one-way scheme can provide high synchronization precision. The

average error of RATS and RITS is in the scale of  $\mu s$  [23].

Bychkovskiy *et al.* [18] propose another one-way scheme implementation. For pair-wise synchronization, the authors use linear regression on collected uni-directional time stamps to compute offset. For multi-hop calibration, they take advantage of existing loops to maximize consistency.

There is an ongoing version of IEEE 1588 [37], or Precision Time Protocol (PTP), which targets at providing networks with clock precision in sub-microsecond. There are two fundamental assumptions that differentiate our works and this effort. First, PTP assumes synchronization messages can be exchanged in a short period of time, such that no relative offset occurs during the exchange process. Second, PTP also assumes the transmit time of synchronization messages between two clocks is a constant. In our works, we do not have this assumptions, thus making our conclusions versatile.

Paxson [31] collected time stamps information from hosts connected by Internet. Due to the high-variance and long latency nature of Internet routing, we can expect their results also apply to the challenging environment specified above. The author propose to use only one-way messages to predict the relative skew, and two-way messages to estimate relative offset between two hosts. Although they have similar solution as ours, detailed theoretical analysis, especially variance analysis and multi-hop performance evaluation, is lacking in this paper.

Ellingson *et al.*'s [35] work is the closest one to our studies. The authors analyze one-way and two-way and propose variance analysis of one-way pair-wise error under the case of dynamic network setting. In addition, the authors propose a recursive algorithm to filter out measurement error. Again, the multi-hop, variance, and the impact of message number analysis for pair-wise two-way is lacking. On the hand, we focus more on the precision with the existence of non-filtered measurement error.

An excellent survey [1] reviews some of the state of the art in time synchronization, that are not mentioned here.

Some synchronization works focus on objectives other than precision or power consumption. Cristian [36] proposes a probabilistic method to alleviate the clock reading error. Their solution is using a probabilistic method to determine a clock reading is valid or not, according to the knowledge of the distribution of network delay. In our works, we don't discard clock readings, and use all collected time-stamps for average performance analysis. Clearly, combining Cristian's technique can improve the results presented in our works.

Lamport [33] propose an algorithm to reconstruct the complete ordering of all events taking place in the system, by using knowledge of partial ordering of a subset of events. They apply this concept to do time synchronization, and bound the synchronization error. One of the main assumption of this paper is, if a message

is sent earlier than another message, then it will also be received earlier. This assumption does not apply to acoustic communication system, due to the slow propagation speed of acoustic waves.

Some existing works deal with the synchronization tasks with special assumptions. Lamport *et al.* [34] propose an algorithm that can provide bounded error, in the existence of "two-faced clock". A two-faced clock provide inconsistent clock readings to different users, thus making many synchronization protocols fail.

TSHL [12] is an implementation specifically designed for high propagation delay networks, which is done in a parallel efforts of our works. However, several things are lacking in this paper [12]. First, the authors only address the propagation delay in their computation, and assume all other jitter terms are negligible. This assumption may not be true when the acoustic traveling distance is not long enough, and application layer time stamping is adopted, thus making the applicability limited. Second, the authors use simple one-way and two-way implementations in their analysis, which is an unfair comparison. In addition, the analysis of their TSHL protocol, and the discussion of multi-hop setting are also lacking.

In sum, synchronization protocols designed for wired environment do not take energy overhead into account explicitly. Although protocols designed for RF-based sensor networks do bear power consumption in mind, they suffer from long propagation delay. On the other hand, although some works are capable of providing precise and energy efficient protocols for the challenging environment mentioned above, comprehensive comparison and performance under multi-hop network setting are lacking.

### 3 PAIR-WISE SYNCHRONIZATION ERROR ANALYSIS

#### 3.1 Definitions and Assumptions of Analysis

By "ideal clock", we are referring to a clock that is capable of measuring time in consistent, and "ideal clock reading" is a clock reading given by an ideal clock. By "clock offset", we are referring to the reading difference between two clocks at the same instance. The rate of one clock "drifting away" from an ideal clock or another non-ideal clock is defined as "clock bias", and "relative clock skew", respectively. In this paper, we assume every node is equipped with an affine clock, clock bias  $\{S_i\}$  are not functions of time and iid (identically, independently distributed), and every node is equipped with identical hardware and software settings. In the following we use a random variable  $\Gamma_i$  to denote the *total* delay/jitter taken for packet  $i$ , transmitted from one node to another and is measured by an ideal clock. In addition, we assume  $\{\Gamma_i\}$  are iid, and  $\{\Gamma_i\}$  and  $\{S_i\}$  are independent.

Some possible sources of delays in packet transmission include application layer send-receive times, access time, link layer transmission and reception time, propagation time, interrupt handling time, encoding and decoding

$T_i$	A clock reading (time stamp) generated by a node involved in a synchronization process.
$t_i$	The time measured by the ideal clock corresponding to the clock reading $T_i$ .
$\Gamma_i$	Total delay/jitter corresponding to packet $i$ as measured by the ideal clock.
$S_i$	Bias of clock $i$ (clock skew between clock $i$ and the ideal clock).
$b_i$	Intercept of clock $i$ (clock reading difference between clock $i$ and the ideal clock at time 0).
$O_{t_i}^a$	Actual offset between reference and unsynchronized node at instance $t_i$ .
$O_{t_i}^e$	Estimated value of $O_{t_i}^a$ .
$RO_{t_i \rightarrow t_j}^a$	Actual offset difference between reference and unsynchronized node contributed by period $t_i$ to $t_j$ . $RO_{t_i \rightarrow t_j}^a = O_{t_i}^a - O_{t_j}^a$
$RO_{t_i \rightarrow t_j}^e$	Estimated value of $RO_{t_i \rightarrow t_j}^a$ .
$\varepsilon_{t_i}^j$	Overall error at time $t_i$ , up to level $j$ .

TABLE 1  
Notations

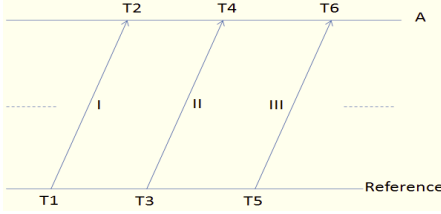


Fig. 1. Message timelines for one-way synchronization scheme

time, and byte alignment time [2]. Note that, even after some delay sources are mitigated through sophisticated time stamping techniques such as those proposed in FTSP [2], the value of  $\Gamma_i, \forall i$  is always positive.

The symbols used in the following contents are listed in Table 1.

### 3.2 Analysis of One-way Dissemination Scheme

An example of one-way scheme is plotted in Figure 1. The arrows in the figure represent directions of transmitted packets for synchronization purposes. The time relations of packet  $I$  can be written as  $T_2 = T_1 + O_{t_1}^{A \rightarrow B} + S_A \Gamma_I$ , and similarly for other packets.

We first define the following matrix:

$$A_M = \begin{pmatrix} 1 & T_1 \\ 1 & T_3 \\ \vdots & \vdots \end{pmatrix}, B_M = \begin{pmatrix} T_2 \\ T_4 \\ \vdots \end{pmatrix}, X_M = \begin{pmatrix} C_M \\ D_M \end{pmatrix} \quad (1)$$

Where matrix  $X_M$  represents the two unknown parameters of doing linear regression on these time stamps. By using linear algebra skills [28], we can compute  $X_M$  by solving  $A_M^T A_M X_M = A_M^T B_M$ . Therefore,

$$X_M = \begin{pmatrix} C_M \\ D_M \end{pmatrix} = (A_M^T A_M)^{-1} A_M^T B_M \quad (2)$$

However, if all uncertainties can be magically eliminated,

matrix  $X_I$  becomes:

$$X_I = \begin{pmatrix} C_I \\ D_I \end{pmatrix} = (A^T A)^{-1} A^T B_I$$

where,  $A = A_M, B_I = B_M - N, N = \begin{pmatrix} S_A \Gamma_I \\ S_A \Gamma_{II} \\ \vdots \end{pmatrix}$  (3)

Thus,  $X_{error} = X_M - X_I = (A^T A)^{-1} A^T N$ . Because:

$$A^T A = \begin{pmatrix} m & \sum_{i=1}^m T_{2i-1} \\ \sum_{i=1}^m T_{2i-1} & \sum_{i=1}^m T_{2i-1}^2 \end{pmatrix}$$

$$A^T N = \begin{pmatrix} S_A \sum_{i=1}^m \Gamma_i \\ S_A (\sum_{i=1}^m T_{2i-1} \Gamma_i) \end{pmatrix}$$

m: number of packets (4)

$X_{error}$  can be calculated as (for brevity purpose, we omit the superscript  $m$  and subscript  $i = 1$  of summation in the following derivation):

$$X_{error} = X_M - X_I = \begin{pmatrix} C_{error} \\ D_{error} \end{pmatrix}$$

$$= \frac{S_A}{m(\sum T_{2i-1}^2) - (\sum T_{2i-1})^2}$$

$$\times \begin{pmatrix} (\sum T_{2i-1}^2)(\sum \Gamma_i) - (\sum T_{2i-1})(\sum T_{2i-1} \Gamma_i) \\ -(\sum T_{2i-1})(\sum \Gamma_i) + m(\sum T_{2i-1} \Gamma_i) \end{pmatrix} \quad (5)$$

Because of independency of  $\{\Gamma_i\}$  and  $\{S_i\}$ , one important observation of equation (5) is that the expected value of  $D_{error}$  equals 0, even the value of  $m$  is the minimal 2. That means, one-way dissemination scheme is capable of estimating clock skew accurately in average, and this result is consonant with Freris et al's conclusion [22].

On the other hand, because  $\{\Gamma_i\}$  are iid, the expected value of  $C_{error}$  becomes  $S \bar{\Gamma}$ , given  $Exp[\Gamma_i] = \bar{\Gamma}$  and  $Exp[S_i] = \bar{S}$ . Since the value of  $\{\Gamma_i\}$  and  $\{S_i\}$  are positive, doing linear regression on time stamps collected from uni-directional packets will over-estimate offset in average, and this conclusion holds no matter how large the value of  $m$  is.

From equation (5), because the variance of  $D_{error}$  is a decreasing function of  $m$ , it would be helpful to use more

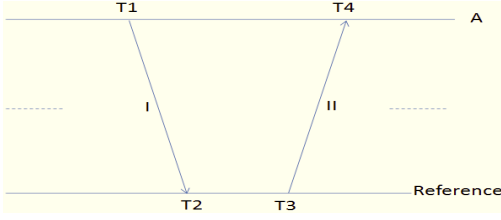


Fig. 2. Message timelines for two-way synchronization schemes

packets if an accurate and low variance skew estimation is desired.

### 3.2.1 An Explanation of FTSP's Performance

In FTSP [2], RITS [11] and RATS [23] we notice the average error of these protocols may fluctuate with respect to the hop distance, i.e. using one-way sometimes may *under-estimate* offset. One possible reason for this confliction is the high variance of  $C_{error}$  and  $D_{error}$ .

Because of the adoption of sophisticated time stamping techniques, e.g. FTSP [2], the value of  $\bar{\Gamma}$  is very small in practice. However, from equation (5), we notice the standard deviation of  $C_{error}$  and  $D_{error}$  is so large comparing with  $\bar{\Gamma}$ . Thus, it is possible to see the situation of under-estimate for one-way scheme if the number of iteration is not large enough. In addition, because the variance increases with respect to hop distance in multi-hop scenario, this phenomenon becomes even more likely when the un-synchronized node is far from the root node.

Even though we do not explicitly model the mechanism of filtering out extreme sample points while doing linear regression in FTSP, this strategy has equivalent effect of using smaller variance of  $\Gamma$  in our analysis. Because we do not make assumption about the variance of  $\Gamma$ , the conclusion of over-estimate clock offset in average still subsists.

### 3.3 Analysis of Two-way Exchange Scheme

Figure 2 is an example of two-way scheme. Similar as prior analysis, we can write:

$$\begin{aligned} T_2 &= T_1 - O_{t1}^a + S_{ref}\Gamma_I \\ T_4 &= T_3 + O_{t3}^a + S_A\Gamma_{II} \end{aligned} \quad (6)$$

From TPSN [3], LTS [13] and Paxson *et al.* [31], the offset estimation is  $\frac{(T_4 - T_3) + (T_1 - T_2)}{2}$ . Because  $O_{t1}^a = O_{t3}^a + (S_{ref} - S_A)(t_3 - t_1)$ , from equation (6):

$$\begin{aligned} O_{t4}^e &= \frac{(T_4 - T_3) + (T_1 - T_2)}{2} \\ &= \frac{1}{2}[(O_{t3}^a + S_A\Gamma_{II}) + (O_{t1}^a - S_{ref}\Gamma_I)] \\ &= O_{t3}^a + \frac{1}{2}[(S_{ref} - S_A)(t_3 - t_1) + (S_A\Gamma_{II} - S_{ref}\Gamma_I)] \end{aligned} \quad (7)$$

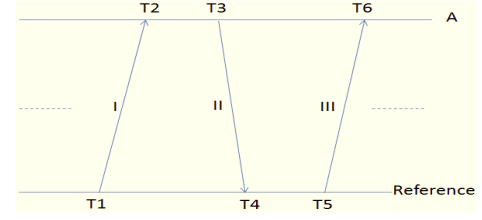


Fig. 3. Message timelines for hybrid synchronization schemes

However,  $O_{t4}^a = O_{t3}^a - (S_{ref} - S_A)(t_4 - t_3) = O_{t3}^a - (S_{ref} - S_A)\Gamma_{II}$ . Thus,

$$\begin{aligned} Error &= (T_4 - O_{t4}^e) - (T_4 - O_{t4}^a) = O_{t4}^a - O_{t4}^e \\ &= -\frac{1}{2}(S_{ref} - S_A)(t_4 - t_1) - \frac{S_{ref}}{2}(\Gamma_{II} - \Gamma_I) \end{aligned} \quad (8)$$

Because the clock skew is difference, the first term in equation (8) represents the relative offset contributed by the period of synchronization process, which we call it "actual relative offset from  $t_1$  to  $t_4$ " and is denoted by  $RO_{t1 \rightarrow t4}^a$ . Since  $\{\Gamma_i\}$  are iid, and  $\{S_i\}$  are iid, the expected error is 0. Therefore, two-way is capable of estimating offset precisely in average. One potential problem of two-way scheme is the high variance of precision. From equation (8), the first term is a function of  $t_4 - t_1$ . Because the variance of this error can be computed as:

$$Variance\ of\ error = \frac{(t_4 - t_1)^2}{2}var[S] + \frac{1}{2}var[ST] \quad (9)$$

If  $(t_4 - t_1)$  is large, then the variance of synchronization precision can be potentially high. Therefore, using two-way scheme in a low-duty, high propagation delay environment may not be able to provide satisfying synchronization precision.

### 3.4 Analysis of Hybrid Scheme

The basic idea of hybrid is: using two-way scheme to estimate offset, and then using the skew estimation calculated from one-way scheme to further refine the offset estimation.<sup>1</sup> An example of simplest 3-packet exchange scenario of hybrid scheme is shown in Figure 3. First, we can write:

$$\begin{aligned} T_2 &= T_1 - O_{t1}^a + S_A\Gamma_I \\ T_4 &= T_3 + O_{t3}^a + S_{ref}\Gamma_{II} \\ T_6 &= T_5 - O_{t5}^a + S_A\Gamma_{III} \end{aligned} \quad (10)$$

From packet *II* and *III* we can compute  $O_{t6}^e = \frac{(T_6 - T_5) + (T_3 - T_4)}{2}$ . From equation (8), we know the error of  $O_{t6}^e$  is  $\frac{1}{2}(S_{ref} - S_A)(t_6 - t_3) + \frac{S_{ref}}{2}(\Gamma_{III} - \Gamma_{II})$ . To

1. TSHL uses one-way skew estimation to calibrate time stamps of un-synchronized node before it proceeds to use two-way scheme to estimate offset. In hybrid scheme, we *ONLY* use the skew estimation to compensate the relative offset taking place during the synchronization process. Therefore, hybrid scheme can operate with only 3 synchronization packets, while TSHL needs at least 4. In addition, hybrid scheme is lighter in terms of computation demand.

reduce the variance contributed by  $RO_{t_3 \rightarrow t_6}^a = \frac{1}{2}(S_{ref} - S_A)(t_6 - t_3)$ , we use the skew estimation calculated from packet I and III to compensate it.

Because the skew estimation is  $\frac{T_5 - T_1}{T_6 - T_2}$ , the estimation of  $RO_{t_3 \rightarrow t_6}^a$  becomes:

$$RO_{t_3 \rightarrow t_6}^e = \frac{T_6 - T_3}{T_6 - T_2}(T_5 - T_1) - (T_6 - T_3) \quad (11)$$

According to the affine clock model assumption, equation (11) can be re-written as:

$$\begin{aligned} RO_{t_3 \rightarrow t_6}^e &= \frac{T_6 - T_3}{T_6 - T_2}(T_5 - T_1) - (T_6 - T_3) \\ &= S_{ref}(t_5 - t_1) \frac{T_6 - T_3}{T_6 - T_2} - S_A(t_6 - t_3) \end{aligned} \quad (12)$$

Thus, the error of hybrid can be calculated as:

$$\begin{aligned} O_{t_6}^e - O_{t_6}^a &= \text{error of } RO_{t_3 \rightarrow t_6}^e + \frac{S_{ref}}{2}(\Gamma_{II} - \Gamma_I) \\ &= \frac{S_{ref}}{2} \left[ \frac{T_6 - T_3}{T_6 - T_2}(\Gamma_{III} - \Gamma_I) + (\Gamma_{II} - \Gamma_I) \right] \end{aligned} \quad (13)$$

Because  $\{\Gamma_i\}$  are iid, the mean error of hybrid equals to 0. In addition, the variance of hybrid scheme is:

$$\begin{aligned} \text{Variance of error} &= \text{var}[S\Gamma] \times \\ &\left( \frac{1}{4} \left( \frac{T_6 - T_3}{T_6 - T_2} \right)^2 + \frac{1}{4} \left( \frac{T_6 - T_3}{T_6 - T_2} + 1 \right)^2 + \frac{1}{4} \right) \end{aligned} \quad (14)$$

Because  $\frac{T_6 - T_3}{T_6 - T_2} < 1$ , we notice the high variance problem of two-way has been mitigated by compensating the relative offset. Thus, hybrid scheme not only has 0 mean error, its variance is also significantly smaller than two-way scheme. Although the hybrid scheme we analyze here is the case with least packet exchange, it is possible to use more uni-directional packets from reference to A, to improve the precision of clock skew estimation. Note that, more uni-directional packets can produce higher skew estimation precision in the expense of higher energy overhead. On the other hand, higher clock skew estimation in a single synchronization instance leads to less frequent re-synch process, given a desired error requirement. Thus, one of our future work is to study the trade-off between number of uni-directional packets involved in a single synchronization process and synchronization precision. We also have interest in finding the optimal number of packets used in hybrid scheme, in terms of network lifespan.

### 3.5 Analysis of Receiver-Receiver Scheme

An example of synchronization packet exchange scenario by using receiver-receiver scheme is shown in Figure 4. Similarly, we can write:

$$\begin{aligned} T_2 &= T_1 + O_{t_1}^{a,ref \rightarrow A} + S_A \Gamma_I \\ T_3 &= T_1 + O_{t_1}^{a,ref \rightarrow B} + S_B \Gamma_{II} \end{aligned} \quad (15)$$

For clarification, we use  $O_{t_1}^{a,ref \rightarrow A}$  to denote the actual offset between reference and A at instance  $t_1$ , and similarly for  $O_{t_1}^{a,ref \rightarrow B}$ . Because the estimated offset is  $T_3 - T_2$ ,

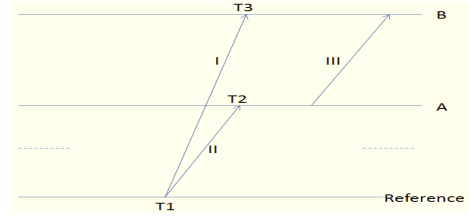


Fig. 4. Message timelines for hybrid synchronization schemes

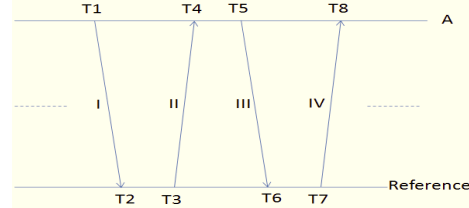


Fig. 5. Message timelines for hybrid synchronization schemes

the error of this offset estimation becomes:

$$\begin{aligned} \text{Error} &= (T_3 - T_2) - (O_{t_1}^{a,ref \rightarrow B} - O_{t_1}^{a,ref \rightarrow A}) \\ &= S_B \Gamma_{II} - S_A \Gamma_I \end{aligned} \quad (16)$$

One potential problem of receiver-receiver scheme is the difference of acoustic traveling distance between reference to A and reference to B. If the difference of traveling distance for A and B are significant, then the error of offset estimation using receiver-receiver scheme can be non-negligible. In addition, receiver-receiver scheme requires A and B to be located within the broadcast region of reference. However, we focus on a tree-structured network with only one node in each level. Thus, receiver-receiver scheme does not fit into our need.

### 3.6 Analysis of Multiple Two-way Exchange Scheme

CTP [24], Tiny-sync, and Mini-Sync [14] are representative works of this scheme. Basically, multiple two-way exchange requires multiple rounds of two-way packet exchange, and then using linear regression to estimate clock offset and skew. An example of multiple two-way exchange scheme is shown in Figure 5. We can write:

$$A_M = \begin{pmatrix} 1 & T_2 \\ 1 & T_3 \\ 1 & T_6 \\ 1 & T_7 \\ \vdots & \vdots \end{pmatrix}, B_M = \begin{pmatrix} T_1 \\ T_4 \\ T_5 \\ T_8 \\ \vdots \end{pmatrix}, X_M = \begin{pmatrix} C_M \\ D_M \end{pmatrix} \quad (17)$$

Similarly as one-way, we can calculate the error as:

$$\begin{aligned} \text{Error} &= X_I - X_M = \frac{S_A}{m(\sum r_i^2) - (\sum r_i)^2} \times \\ &\begin{pmatrix} (\sum r_i^2)(\sum (-1)^{i+1} \Gamma_i) - (\sum r_i)(\sum r_i(-1)^{i+1} \Gamma_i) \\ -(\sum r_i)(\sum (-1)^{i+1} \Gamma_i) + m(\sum r_i(-1)^{i+1} \Gamma_i) \end{pmatrix} \end{aligned} \quad (18)$$



Where  $\{r_i\}$  represents the time stamps generated by reference, i.e.  $T_2, T_3, T_6, T_7 \dots$  etc. One observation is that the expected value of  $C_{error}$  is strictly larger than 0, and the mean value of  $D_{error}$  is strictly less than 0. Obviously, this is not an ideal solution comparing with hybrid scheme.

## 4 ANALYSIS OF SYNCHRONIZATION ERROR UNDER MULTI-HOP SITUATIONS

### 4.1 Definitions and Assumptions of Analysis

Because of the existence of clock skew, a synchronized clock, say A, may already significantly drift away from the reference clock before another clock, say B, tries to synchronize with it. If clock A uses skew estimation to calibrate itself before providing readings to other clocks, the inaccuracy of clock skew estimation still deteriorates the synchronization precision. Therefore, the relative offset taking place during the synchronization process is not a negligible error source when synchronization process is long. In multi-hop networks with sleep-scheduling, which can increase the intervals between two consecutive synchronization process, this source of error is significant. We call this error as “relative offset error”. Intuitively, if precise clock skew estimation is possible, then we can benefit from compensating relative offset error; otherwise, it may further deteriorate synchronization precision. We analyze both cases separately in the following sections. The used symbols are listed in Table 1, and we make the following assumptions:

- 1) We assume a tree-structured network, and every node is assigned to a specific level, determined by its hop distance to the root/reference node. The root node is assigned to level 0, and synchronization takes place sequentially from the root to higher level node.
- 2) Every synchronization instance is independent with each other.
- 3) By “overall error up to level  $i$ ”, we refer to the clock reading difference between calibrated level  $i$  clock and the root clock, at the instance when level  $i$  node completes its synchronization with level  $i-1$  node.

### 4.2 Propagation Error Without Relative Offset Compensation

#### 4.2.1 One-way Dissemination Scheme

An example scenario is plotted in Figure 6. Similarly, we can compute  $X_{M,i+2} = (C_{M,i+2}, D_{M,i+2})^T = [(A_{i+2} + P_{i+2})^T (A_{i+2} + P_{i+2})]^{-1} (A_{i+2} + P_{i+2})^T B_{M,i+2}$ , where:

$$A_{i+2} = \begin{pmatrix} 1 & T_7 \\ 1 & T_9 \\ \vdots & \vdots \end{pmatrix}, P_{i+2} = \begin{pmatrix} 0 & -O_{t_6, i+1}^e \\ 0 & -O_{t_6, i+1}^e \\ \vdots & \vdots \end{pmatrix},$$

$$B_{M,i+2} = \begin{pmatrix} T_8 \\ T_{10} \\ \vdots \end{pmatrix} \quad (19)$$

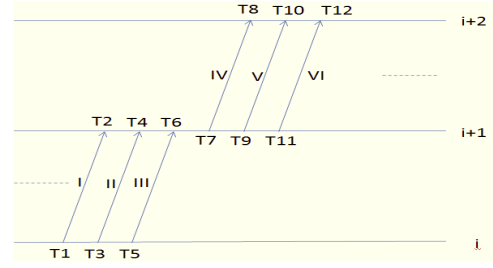


Fig. 6. An example scenario for doing synchronization in a multihop network using one-way scheme.

For clarification, we add the subscripts  $i+2$  to  $A_{i+2}$ ,  $P_{i+2}$  and  $B_{M,i+2}$  to identify these matrixes/vlaues are computed between level  $i+2$  and  $i+1$  clock, and add the subscript  $i+1$  in  $O_{t_6, i+1}^e$  to represent this offset estimation is computed between level  $i+1$  and  $i$  clocks. Then, the error matrix  $X_{error, i+2}$  can be computed as:

$$X_{error, i+2} = \frac{S_{i+2}}{m \sum (T_{2i-1}^e)^2 - (\sum T_{2i-1}^e)^2} \times \begin{pmatrix} (\sum (T_{2i-1}^e)^2)(\sum \Gamma_i) - (\sum T_{2i-1}^e)(\sum (T_{2i-1}^e \Gamma_i)) \\ -(\sum T_{2i-1}^e)(\sum \Gamma_i) + m(\sum (T_{2i-1}^e \Gamma_i)) \end{pmatrix} \quad (20)$$

For brevity, we use  $T_{2i-1}^e = T_{2i-1} - O_{t_6, i+1}^e$  to represent the clock reading  $T_{2i-1}$  from level  $i+1$ , such that it has been calibrated by using offset estimations between level  $i+1$  and  $i$ . Because  $\{\Gamma_i\}$  are iid, we can compute the expected value of  $X_{error, i+2}$  as:

$$Exp[X_{error, i+2}] = \frac{S_{i+2}}{m \sum (T_{2i-1}^e)^2 - (\sum T_{2i-1}^e)^2} \times \begin{pmatrix} m\bar{\Gamma}(\sum (T_{2i-1}^e)^2) - \bar{\Gamma}(\sum T_{2i-1}^e)^2 \\ -m\bar{\Gamma}(\sum T_{2i-1}^e) + m\bar{\Gamma}(\sum T_{2i-1}^e) \end{pmatrix} \quad (21)$$

Two conclusions can be made by observing equation (21). First, the mean error of  $D_{M, i+2}$  is 0. That means, one-way can provide precise clock skew estimation between  $i+1$  and  $i+2$  in average, without using clock skew estimation. The other observation is that mean error of  $C_{M, i+2}$  equals  $\bar{S\Gamma}$ . Therefore, using one-way without compensating relative offset error will produce excess  $\bar{S\Gamma}$  amounts of intercept estimation error in average. Because the intercept estimation error in level  $i+2$  is on top of prior errors, using one-way without relative offset compensation will lead to unbounded error under multi-hop scenario in average. By induction, we can write:

$$Exp[\text{Overall Error up to level } j] = j\bar{S\Gamma} \quad (22)$$

#### 4.2.2 Two-way Exchange Scheme

An example scenario is plotted in Figure 7. For convenience, we denote the reference clock reading at instance  $t_i$  as  $T_i^a$ , i.e.  $T_i^a = S_0 t_i + b_0$ . By definition, we can write

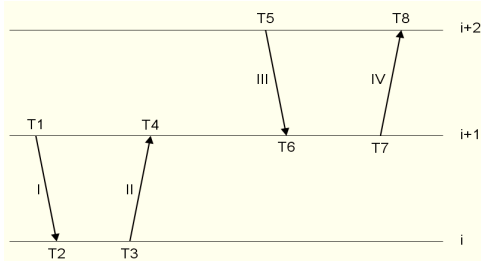


Fig. 7. An example scenario for doing synchronization in a multihop network using two-way scheme.

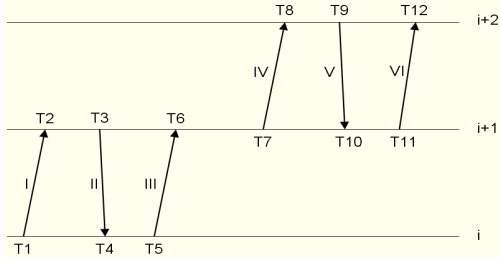


Fig. 8. An example scenario for doing synchronization in a multihop network using hybrid scheme.

$\varepsilon_{t_4}^{i+1} = T_4^e - T_4^a$ . Because:

$$\begin{aligned} T_6^e &= T_6 - O_{t_4, i+1}^e \\ T_7^e &= T_7 - O_{t_4, i+1}^e \\ O_{t_8, i+2}^e &= \frac{(T_8 - T_7^e) + (T_5 - T_6^e)}{2} \end{aligned} \quad (23)$$

Using the affine clock model, e.g.  $T_6 = S_{i+1}t_6 + b_{i+1}$ , we can thus compute  $\varepsilon_{t_8}^{i+2}$  as:

$$\begin{aligned} \varepsilon_{t_8}^{i+2} &= T_8^e - T_8^a = (T_8 - O_{t_8, i+2}^e) - T_8^a \\ &= \varepsilon_{t_4}^{i+1} + S_0(t_4 - t_7 - \Gamma_{IV}) \\ &+ S_{i+1} \frac{t_6 + t_7 - 2t_4}{2} + S_{i+2} \frac{t_7 - t_6 + \Gamma_{III} + \Gamma_{IV}}{2} \end{aligned} \quad (24)$$

Because  $\{S_i\}$  and  $\{\Gamma_i\}$  are iid, the mean value of  $\varepsilon_{t_8}^{i+2}$  equals  $\varepsilon_{t_4}^{i+1}$ . From section 3.3 we know the average error up to level 1 is 0. By induction, we can have:

$$\text{Exp}[\text{overall error at level } j] = 0 \quad (25)$$

However, one potential problem of using two-way without relative offset compensation is the high variance of error. By observing equation 24, because every synchronization task introduces a constant error on top of prior hop's overall error in average, it can be expected using two-way without relative offset error compensation demonstrates linear increasing variance with respect to hop count distance. This result is also consonant with Giridhar and Kumar's result [20]. In addition, if the length of synchronization process and inter-synchronization period are large, we may end up with huge propagation error in some instances.

#### 4.2.3 Hybrid Scheme

An example scenario is plotted in Figure 8. Overall error at  $t_6$  up to level  $i+1$  as  $\varepsilon_{t_6}^{i+1} = T_6^e - T_6^a$ , and the offset estimation at  $t_{12}$  can be computed as:

$$O_{t_{12}, i+2}^e = \frac{(T_{12} - T_{11}^e) + (T_9 - T_{10}^e)}{2} - RO_{t_9 \rightarrow t_{12}}^e \quad (26)$$

Using the affine clock assumption, and the clock skew estimation  $D_{M, i+2}$  derived from doing linear regression on packets sent by level  $i+1$  and received by  $i+2$ , we can compute overall error at  $t_{12}$  up to level  $i+2$  as:

$$\begin{aligned} \varepsilon_{t_{12}}^{i+2} &= \varepsilon_{t_6}^{i+1} + S_{i+2} \frac{t_{12} - t_9}{2} + S_{i+1} \frac{t_{10} + t_{11} - 2t_6}{2} \\ &- S_0(t_{12} - t_6) + (D_{M, i+2} - 1)(t_{12} - t_9) \end{aligned} \quad (27)$$

Since  $\{S_i\}$  are iid, the expected value of  $\varepsilon_{t_{12}}^{i+2}$  equals  $\varepsilon_{t_6}^{i+1}$ . From section 3.4, we know the mean value of overall error up to level 1 is 0. By induction, we know the mean overall error at any level  $j, \forall j \geq 1$  is still 0. Therefore, using hybrid without compensating relative offset error can produce good synchronization precision under multi-hop scenario in average. However, the trade-off of not compensating relative offset error is high variance of precision, and this effect will be reflected by large value and high variance of  $D_{M, i+2}$ . Because hybrid can produce accurate clock skew estimation, using hybrid without compensating relative offset error is clearly a sub-optimal strategy.

### 4.3 Propagation Error With Relative Offset Compensation

#### 4.3.1 One-way Dissemination Scheme

Consider Figure 6 again. The effect of compensating relative offset error makes the matrix  $P_{i+2}$  different from equation (19), thus changing the value of  $\{T_{2i-1}^e\}$  in equation (20) and (21). However, this effect will be canceled out because they both appear in the numerator and denominator of equation (21). Therefore, no matter we compensate relative offset error or not, using one-way will lead to unbounded multi-hop error in average. Since  $\{S_i\}$  and  $\{\Gamma_i\}$  are both iid and independent with each other, by induction, we know the expected overall error up to level of using one-way with relative offset error is the same as equation (22). Although the mean error is the same as not compensating inter-sync error, compensating relative offset error does provide lower variance, due to the fact one-way is capable of estimating clock skew precisely.

#### 4.3.2 Two-way Exchange Scheme

Consider Figure 7 again. Because both TPSN [3] and LTS [13] do not provide skew estimation, we assume its skew estimation by doing linear regression on collected time stamps. Thus, the clock skew estimation can be computed as  $\frac{T_4 - T_1}{T_3^e - T_2^e}$ , where  $T_2^e$  and  $T_3^e$  represent the relative offset calibrated time stamps corresponding to  $T_2$  and  $T_3$ , respectively. For convenience, we denote the



clock skew estimation between level  $j$  and  $j+1$  as  $\alpha_{j+1}^e$ . By using affine clock model, we can compute the error of skew estimation between level  $i$  and  $i+1$  as:

$$\text{Error of } \alpha_{i+1}^e = \frac{S_{i+1}(t_3 - t_2 + \Gamma_I + \Gamma_{II})}{S_i(t_3 - t_2) + RO_{t_2 \rightarrow t_3}^e} \quad (28)$$

Because  $RO_{t_2 \rightarrow t_3}^e = \alpha_i^e(t_3 - t_2)$ , the error of clock skew estimation in one level will further deteriorate clock skew estimation in its next level. By changing the value of  $T_6^e$  and  $T_7^e$  in equation (23), we can calculate the overall error up to level at  $t_8$  as:

$$\begin{aligned} \varepsilon_{t_8}^{i+2} = T_8^e - T_8^a = & \varepsilon_{t_4}^{i+1} + S_{i+1} \frac{t_7 + t_6 - 2t_4}{2\alpha_{i+1}^e} \\ & + \frac{S_0}{2} \left[ \left(1 - \frac{1}{\alpha_{i+1}^e}\right)(t_6 + t_7) - \frac{2t_4}{\alpha_{i+1}^e} - 2t_8 \right] \\ & + S_{i+2} \frac{t_7 - t_6 + \Gamma_{III} + \Gamma_{IV}}{2} \end{aligned} \quad (29)$$

Since  $\{S_i\}$  and  $\{\Gamma_i\}$  are both iid and independent with each other, the expected overall error becomes:

$$\text{Exp}[\varepsilon_{t_8}^{i+2}] = \varepsilon_{t_4}^{i+2} - \bar{S} \frac{2t_4}{\alpha_{i+1}^e} \quad (30)$$

We first notice, offset estimation error in current hop will be carried to the next hop. Second, the skew estimation keeps adding error in every hop and also deteriorates the offset estimation in next hop. Therefore, the overall error of using two-way with compensating relative offset error under multi-hop scenario will keep accumulating quadratically.

### 4.3.3 Hybrid Scheme

Consider Figure 8 again. Due to the affine clock model assumption, the overall error up to level  $i+2$  at time  $t_{12}$  can thus be calculated as:

$$\begin{aligned} \varepsilon_{t_{12}}^{i+2} = & \varepsilon_{t_6}^{i+1} + (S_{i+1} + D_{M,i+1} - 1) \frac{t_{10} + t_{11} - 2t_6}{2} \\ & - S_0(t_{12} - t_6) + S_{i+2} \frac{t_{12} - t_9}{2} + (D_{M,i+2} - 1)(t_{12} - t_9) \end{aligned} \quad (31)$$

Since  $\{S_i\}$  and  $\{\Gamma_i\}$  are both iid and independent with each other, the expected value of  $\varepsilon_{t_{12}}^{i+2}$  becomes  $\varepsilon_{t_6}^{i+1}$ . In addition, because we know the average overall error up to level 1 is 0 from Section 3.4, by induction, we can write:

$$\text{Exp}[\text{overall error up to level } j] = 0, \forall j \quad (32)$$

Again, since every synchronization task adds in a constant error on top of prior hop's propagation error in average, it can be expected the variance of using hybrid with relative offset compensation increases linearly with respect to level number.

## 5 EXPERIMENTS

Our focus in this paper is on evaluating different schemes for high propagation delay environments, such as underwater acoustic sensor networks. However, we are not aware of any acoustic underwater testbeds that

would be suitable for experimental evaluations. Instead, we use traces collected from experiments with RF-based sensor nodes (specifically the Moteiv Tmote Sky platform [29]). State of the art RF-based time synchronization mechanisms such as FTSP and TPSN advocate the use of MAC-layer time stamping to reduce communication jitter. However, our testbed uses application-layer time-stamping, which incurs a higher latency and variance from sender to receiver. This sufficiently mimics the high propagation delay and high variance characteristics of interest to us. Although underwater acoustic sensor networks can see propagation delays on the order of hundreds of milliseconds, while the application layer time stamping on our testbed gives per-hop latencies of around ten milliseconds, it still suffices to show the relative strengths and weaknesses of the various techniques.

We have chosen to evaluate various schemes by first collecting time-stamp data traces from a common set of experiments and then undertaking offline simulations, instead of directly implementing these schemes on the motes and running them separately online. There are several good reasons for adopting this approach. First, the offline method will yield essentially the same performance as an online implementation as the only key difference is in where the computations take place (on-mote versus off-board on a PC). Since the calculations involved in the different schemes that we evaluate (one-way, two-way, and hybrid) are extremely lightweight, the effect of additional hardware capability on performance is negligible. We instead gain several benefits from this approach. The testbed experiments we use are set up in such a way (all nodes can hear packets from each other) that they can be easily used to generate simulations of  $n!$  permutations of multihop chains. This allows us to average results over a thousand runs trivially, to generate results with high statistical significance. By contrast, if the algorithms were to be run online, we would have to manually change the topology a thousand times while doing the experiments, which would make it prohibitively time-consuming to achieve the same significance. Further, this approach is inherently fair to the various schemes, as they are each evaluated over the same set of time-stamp data. Finally, the traces from the experiments that we undertook in order to obtain these simulations can be reused in the future as a benchmark for other techniques and mechanisms; to this end, we have made our experimental traces publicly available [30].

We now detail the experiments, and then present and comment on the results obtained.

### 5.1 Experimental Setup for Collection of Time-Stamp Traces

We use 20 motes and mark each of them with a unique physical ID number, from 1 to 20. Every mote is placed within the transmission range of others, thus, each mote is capable of communicating directly with any other one in our experiment.

A synchronization packet includes the following information: Round Number, Sender Time Stamp, and Receiver Time Stamp, which are explained as follows:

- 1) Round Number: A round is consisted of a sequential synchronization packets sent through Node ID 1 to Node ID 20. Each round consists of 20 sending cycles. (Detailed in following contents.)
- 2) Time Stamp: The clock readings when a synchronization packet is constructed.

Our data collection experiment proceeds as follows:

- 1) Exactly one node is scheduled to transmit packets in any instance, and all the others are considered as receivers. At time 0, ID 1 node starts broadcasting ten synchronization packets at an interval of 10 *ms*. Once it finishes transmitting 10 packets, a sending cycle is complete.
- 2) Whenever a receiver hears a synchronization packet, it records the sending time stamp, sender ID, and the corresponding local receive time stamp for that packet. Only the first packet, which is commonly received by all receivers, is kept for future use and the remaining packets within the same sending cycle are discarded. Because we need to guarantee the existence of such packet in one sending cycle despite possible transmission errors, we make each sender broadcast 10 packets within its sending cycle.
- 3) Nodes in the network initiate their own sending cycle in sequence of their physical ID. Because every sending cycle takes roughly 100 *ms*, we make the node with physical ID  $i + 1$  start its cycle after 150 *ms* of the first received packet from ID  $i$  node to avoid overlapping prior sending cycle.
- 4) When the node with ID 20 finishes its sending cycle, one sending round completes. To avoid overlapping of two sending rounds, ID 1 node waits one minute and start a new sending round, after it receives packets sent by ID 20 node.
- 5) Repeat the above steps. In our experiments, we have collected 80 sending rounds of data.

## 5.2 Multi-hop Error Calculation

By choosing a permutation of the 20 nodes in our experiments, we obtain a virtual 19-hop line network with exactly one node in each level. The node which provides the reference clock for other nodes is located in level 0. The sequence of time-stamped message exchanges for each scheme are obtained by using the packet from each round of the experimental trace. To simulate a different network instance, we simply pick a different permutation of the 20 nodes. This allows us to obtain statistically meaningful results by averaging over different independent random network.

For each permutation, we apply the different synchronization schemes (one-way, two-way and hybrid; with and without relative offset compensation) offline using Matlab and compute the error at each level as follows.

Synchronization is conducted by calibrating the clocks sequentially level-by-level from 0 to 19, with each scheme. Error at level  $i$  is computed by obtaining the clock difference between “calibrated level  $i$  clock reading” and “the corresponding level 0 clock reading at the same instance”. By “level 0 clock reading at the same instance”, we are referring to the receive time of the last packet involved in level  $i$ ’s synchronization process, measured by the node assigned to level 0 in the same permutation<sup>2</sup>. Presented results for each technique are average of error at each level across 1000 permutations.

## 6 RESULTS AND DISCUSSION

### 6.1 Predicted outcomes

In the trace based simulations, we have the following conditions which are relevant to our analysis models:

- The time periods between two consecutive packets, within the same synchronization process, are approximately the same.
- The time periods between two consecutive synchronization process, i.e. the duration between the end of a synchronization process and the beginning of the next immediate process, are approximately the same.
- The above two values are approximately the same.

By applying above conditions, the predicted outcomes of our experiments are listed in Table 2 and 3. The former adopts skew estimation to compensate for relative offset error, whereas the later does not.

### 6.2 Experiment Results and Discussion

The average 19-hop propagation error of all three synchronization schemes with and without inter-sync error compensation are listed in Table 2 and Table 3, respectively, and are plotted in Figure 9 and Figure 10, respectively. The outcomes presented are averaged over 1000 synchronization iterations. For comparison purpose, we list the results side by side to the predicted outcomes.

As listed in Table 2, the hybrid scheme performs the lowest average error, while two-way scheme performs the highest. A more interesting conclusion can be made by observing Figure 9. One-way scheme maintains a linearly increasing trend with respect to the hop distance to the root node, and two-way scheme raises quadratically. Both schemes suffer from unbounded error. On the other hand, hybrid scheme not only performs significantly better precision comparing to the other two schemes under the same settings, most importantly, hybrid scheme

2. In other words, we use a receiver-receiver scheme to determine the error. It is for sure this method may introduce some “noise” into the actual error. However, the average error introduced by this mechanism is insignificant. According to our analysis in section 3.5, the major problem of using receiver-receiver scheme is the difference of  $\Gamma_I$  and  $\Gamma_{II}$ . Because every T-mote in our experiment is configured identically, and physically located nearby, it can be expected  $\Gamma_I$  and  $\Gamma_{II}$  has approximately the same distribution. Thus, the average error introduced by this method is negligible.

	One-Way	Two-Way	Hybrid
Measured Average Overall Propagation Error	176 <i>ms</i>	4535 <i>ms</i>	-0.18 <i>ms</i>
Predicted Average Overall Propagation Error ( $\varepsilon^{19}$ )	$19 \times (E[\Gamma])$	$510 \times (E[\Gamma])$	$\approx 0$
Measured Average Per-hop Propagation Error	9.26 <i>ms</i>	238.7 <i>ms</i>	-0.095 <i>ms</i>
Estimated Average Per-hop Propagation Error ( $\varepsilon^{19}/19$ )	$E[\Gamma]$	$26.8 \times (E[\Gamma])$	$\approx 0$

TABLE 2

Propagation error under different synchronization schemes with relative offset error compensation

	One-Way	Two-Way	Hybrid
Measured Average Overall Propagation Error	167.4 <i>ms</i>	- 0.2 <i>ms</i>	0.38 <i>ms</i>
Predicted Average Overall Propagation Error ( $\varepsilon^{19}$ )	$19 \times (E[\Gamma])$	$\approx 0$	$\approx 0$
Measured Average Per-hop Propagation Error	8.8 <i>ms</i>	- 0.0105 <i>ms</i>	0.02 <i>ms</i>
Predicted Average Per-hop Propagation Error ( $\varepsilon^{19}/19$ )	$E[\Gamma]$	$\approx 0$	$\approx 0$

TABLE 3

Propagation error under different synchronization schemes without relative offset error compensation

is capable of providing bounded average propagation error.

By comparing the one-way scheme with our predicted value, the mean value of  $\Gamma$  thus can be computed as  $167ms/19 \approx 8.8ms$ . Similarly, the average value of  $\Gamma$  calculated from two-way scheme can be computed as  $4535ms/510 \approx 8.9ms$ , which is very close to what we get from one-way calculation. For the hybrid case, while the predicted value is 0, our experiment outcomes show fluctuated propagation error at different hop distance. The reason of this unmatched results is caused by the high jitter nature of  $\Gamma$ . From the traces, we can calculate the mean of  $\Gamma$  is around 8.9 *ms*, the standard deviation is 2.3, and, in addition, the distribution of  $\Gamma$  is very close to normal distribution. In both one-way and two-way scheme, since the mean propagation error is large, the fluctuation caused by the variance is not obvious at all. However, the hybrid scheme is very precise such that little fluctuations will be easily observed. Based on the above discussions, all these experiment outcomes are consistent with our analysis conclusions.

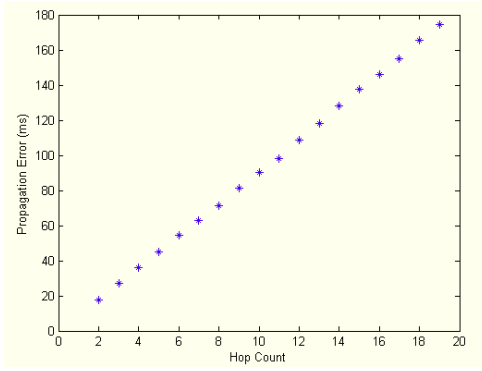
In Table 3, although one-way and hybrid scheme still performs similarly as they do in previous setting, two-way scheme possesses considerably lower propagation error comparing with the prior settings. Especially, two-way scheme, as hybrid scheme, shows no obvious trend of increasing propagation while hop distance goes up in Figure 10, and the synchronization precision of both two-way and hybrid schemes are comparable. These results also match our prediction. From the one-way side, because of the precise clock skew estimation, most of the relative offset error can be mitigated efficiently. Thus only the offset estimation error will propagate hop by hop. By calculating the average value of  $E[\Gamma]$  from the experiment outcomes, we get  $171ms/19 \approx 8.9ms$ , which is the same as prior value. For the hybrid scheme, because it is capable of providing accurate offset and skew estimations simultaneously, the results presented also match our expectation. For the two-way scheme, due to the precise offset estimation we can get from it,

the dominant error source is the relative offset taken place during inter and intra synchronization period. However, because of the equal probability that the clock bias of one node is faster or slower than the other clock that intends to synchronize with itself, it can be expected that the relative offset are equally likely to get positive and negative values. In addition, the value of  $T_{inter}$  and  $T_{intra}$  is a constant in our trace based simulations, it is also reasonable to expect the relative offset will cancel each other throughout different synchronization iterations. A more detailed descriptions can be found in Section 4.

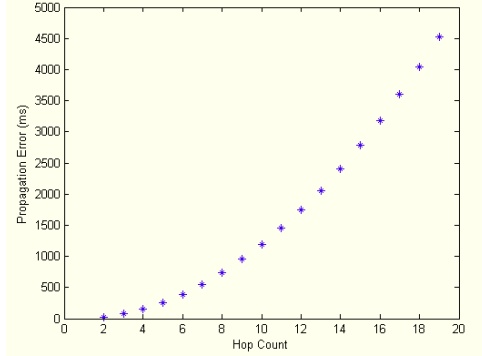
From the above results, we notice while the average propagation error of one-way increases without bounded, two-way without inter-sync error compensation and our hybrid scheme (no matter compensating inter-sync error or not) perform comparably. In addition, both the later two strategies provide bounded propagation error in average. To further address the benefits of hybrid scheme, we compare the variance of these two strategies in the following section.

### 6.3 Variance Comparison Between Hybrid and Two-way Scheme

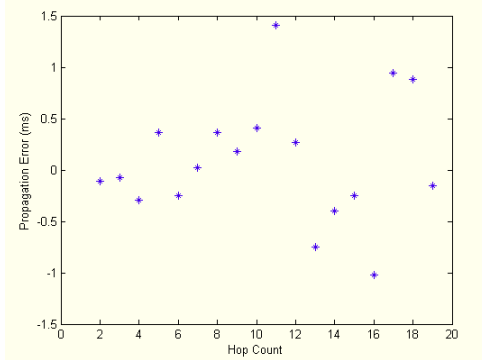
For comparison purpose, we compare the variance of precision between hybrid and two-way scheme under two different strategies: with or without relative offset error compensation. The results are plotted in Figure 11(a) and 11(b). In sum, both hybrid and two-way schemes perform linearly increasing trend with respect to hop distance to the root node. This conclusion is also consonant with Giridhar and Kumar's work [20]. By comparing 11(a) and 11(b) we notice that, hybrid does benefit from compensating relative offset error. However, the variance performance of hybrid is not impressive, considering the extra packets exchange it costs. One possible reason is, the relative offset error is not significant. In our time stamps collecting experiment, since all motes are placed near to each other, the environment may not cause high enough skew between different clocks. In



(a) One-way



(b) Two-way

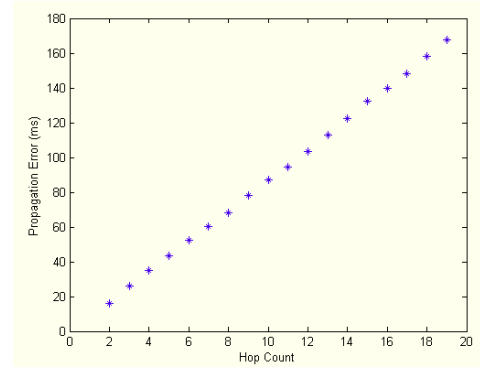


(c) Hybrid

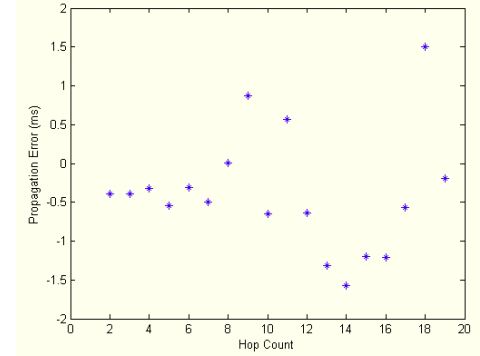
Fig. 9. Trace based simulation results showing multi-hop error propagation for one-way, two-way and hybrid schemes with inter-sync error compensation.

addition, the length of a synchronization process and the inter-synchronization period are not long enough, either. Therefore, the relative offset is not a significant source of error. Instead, the high variance of  $\{\Gamma_i\}$  plays the major role on determining the variance of both schemes. In order to verify our speculation, we intentionally increase the inter-synchronization period to around 12 minutes and re-do the trace-based simulation to simulate a 6-node, 5-hop line network. The variance comparison are plotted in Figure 11(c) and 11(d).

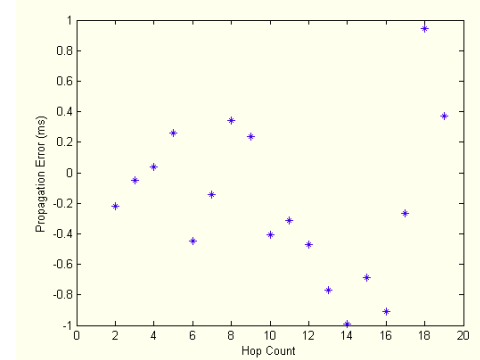
One interesting observation is, variance of two-way without relative offset compensation increases much faster than prior setting. Even the 5-hop variance is higher than prior result of 20-hop. Although the



(a) One-way



(b) Two-way



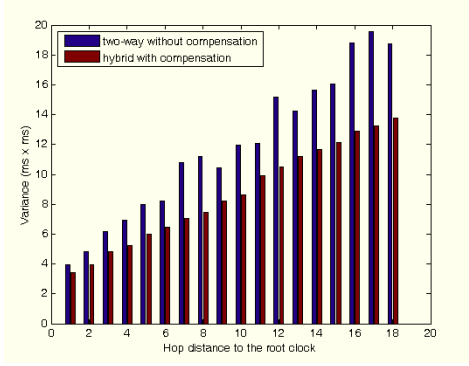
(c) Hybrid

Fig. 10. Trace based simulation results showing multi-hop error propagation for one-way, two-way and hybrid schemes without inter-sync error compensation.

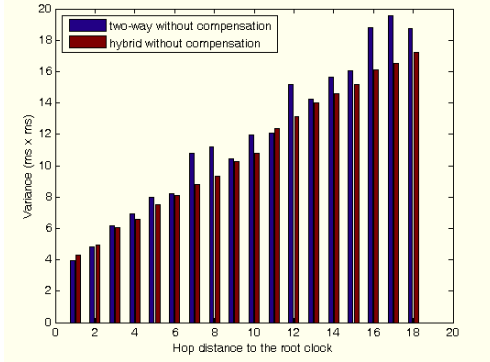
variance of hybrid also increases with longer inter-synchronization period, the rate is much slower. By comparing Figure 11(c) and 11(d), we also notice that hybrid does benefit from compensating relative offset error. From above results, hybrid with relative offset compensation does perform more gracefully than other existing strategies in a high propagation delay and low-duty cycle network.

## 7 SUMMARY AND FUTURE WORK

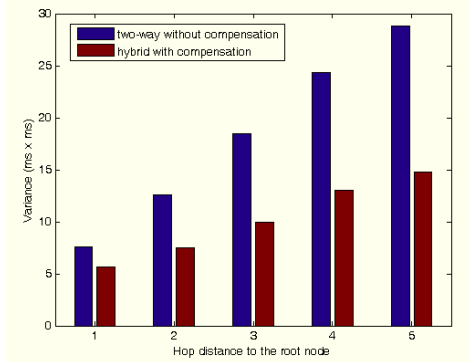
The major contributions of this work are as follows. First, we have presented a thorough comparison of two-way packet exchange, one-way dissemination schemes, receiver-receiver scheme, and multiple two-way scheme.



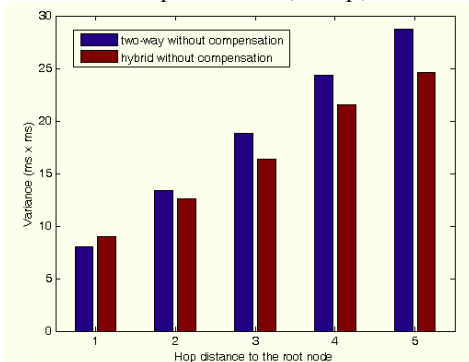
(a) Two-way without compensation vs. hybrid with compensation (19-hop)



(b) Two-way without compensation vs. hybrid without compensation (19-hop)



(c) Two-way without compensation vs. hybrid with compensation (5-hop)



(d) Two-way without compensation vs. hybrid without compensation (5-hop)

Fig. 11. Variance comparison between two-way and hybrid scheme.

We find that, one-way dissemination performs worse in estimating clock offset, two-way dissemination performs worse while relative offset error is significant, receiver-receiver scheme performs worse when propagation delay is substantially different, and multiple two-way performs biased offset and skew estimation. Second, we have proposed a hybrid one-way dissemination/two-way exchange synchronization scheme that can provide substantially better accuracy, bounded error propagation over multiple hops in average, and low variance of propagation error. Such a strategy would be most useful for a challenging environment, characterized by high propagation delay, low-duty cycle, and highly skewed clocks, such as underwater acoustic sensor networks. Third, we implement a series of experiments to collect time stamps from real Tmotes. From the trace-based simulation results, we get the same conclusions as we make in our analysis.

In the future, we hope to extend this work by doing fine-grained time stamping in practical experiments, and develop a time synchronization protocol based on the hybrid scheme that will be suitable for multi-hop settings. It would then be of great value to compare the different approaches in detail through real implementations on wireless devices. In addition, we would like to find out the optimal number of packets used in hybrid scheme, in terms of maximum network lifespan.

## 8 ACKNOWLEDGEMENTS

Thanks our anonymous reviewers as well as members of the USC Autonomous Networks Research Group for their constructive and valuable feedback on this work.

## REFERENCES

- [1] B. Sundararaman, U. Buy, A.D. Kshemkalyani, "Clock Synchronization in Wireless Sensor Networks: A Survey", *Ad Hoc Networks*, 3(3): 281-323, May 2005.
- [2] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The Flooding Time Synchronization Protocol," *SenSys*, November 3-5, 2004, Baltimore, Maryland, USA.
- [3] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-sync Protocol for Sensor Networks," *SenSys*, November 5-7, 2003, Los Angeles, California, USA.
- [4] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," in Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002.
- [5] P. Huang and B. Krishnamachari, "Analysis of existing approaches and a new hybrid strategy for synchronization in sensor networks," in Proceedings of the Fifth (EmNets 2006), Boston, MA, May 2006.
- [6] S. Ganeriwal, D. Ganesan, M. Hansen, M. B. Srivastava, D. Estrin, "Rate-adaptive time synchronization for long-lived sensor networks," Proceedings of ACM SIGMETRICS international conference on measurement and modeling in computer, June 2005.
- [7] R. Karp, J. Elson, D. Estrin, S. Shenker, "Optimal and Global Time Synchronization in Sensor networks," UCLA CENS Technical Report 0012, April 10, 2003.
- [8] J. Elson, K. Romer, "Wireless Sensor Networks: A New Regime for Time Synchronization," Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I), Princeton, New Jersey. October 2002.
- [9] F. Sivrikaya, B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, July-Aug. 2004, Vol. 18, Issue 4, page 45-50.

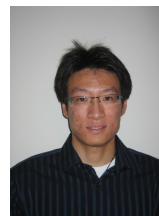
- [10] F. Zhang, G. Deng, "Probabilistic time synchronization in wireless sensor networks," *Proceedings of IEEE Wireless Communications, Networking and Mobile Computing*, 2005.
- [11] J. Sallai, B. Kusy, A. Ledeczi, P. Dutta, "On the scalability of routing integrated time synchronization," *European Workshop on Wireless Sensor Networks (EWSN 22006)*, Zurich, Switzerland, February 13, 2006.
- [12] A. Syed, J. Heidemann, "Time Synchronization for high latency acoustic networks," *Proceeding of Infocom*, 2006.
- [13] J.V. Greunen, J. Rabaey, "Lightweight time synchronization for sensor networks," *Proceeding of ACM WSNA*, September 2003.
- [14] M.L. Sichitiu, C. Veerarittiphan, "Simple, accurate Time Synchronization for wireless sensor networks," *Proceeding of IEEE conference on wireless communications and networking (WCNC)*, March 2003.
- [15] S. Ganeriwal, D. Ganesan, H. Sim, V. Tsiatsis, M.B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," *Proceeding of Sensys*, November 2005.
- [16] J. Elson, D. Estrin, "Time synchronization for wireless sensor networks," *UCLA CS technical report 200028*, 2001.
- [17] A. Hu, S.D. Servetto, "Algorithmic aspects of the time synchronization problem in large-scale sensor networks," *ACM/Kluwer Journal on Mobile Networks and Applications (MONET)*, Special Issue on Wireless Sensor Networks, 2003.
- [18] V. Bychkovskiy, S. Megerian, D. Estrin, M. Potkonjak "A collaborative approach to in-place sensor calibration," *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, volume 2634 of *Lecture Notes in Computer Science*, pages 301-316, Springer-Verlag Inc., 2003.
- [19] R.M. Karp, J. Elson, C.H. Papadimitriou, S. Shenker, "Optimal and Global Time Synchronization in Sensornets," *CENS Technical Report 0012*, April 10, 2003.
- [20] A. Giridhar, P.R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 4915-4920, San Diego, Dec. 13-15, 2006.
- [21] R. Solis, V.S. Borkar, P.R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2734-2739, San Diego, Dec. 13-15, 2006.
- [22] N.M. Freris, P.R. Kumar, "Fundamental limits on synchronization of affine clocks in networks," *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Dec. 12-14, 2007.
- [23] B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, D. Culler, "Elapsed time on arrival: a simple and versatile primitive for canonical time synchronization services," *Int. J. Ad Hoc ubiquitous computing*, 2006.
- [24] S. Graham, P.R. Kumar, "Time in general-purpose control systems: the control time protocol and an experimental evaluation," *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 4004-4009, Bahamas, Dec. 14-17, 2004.
- [25] K. Shin, K. Lee, K. Lee, P. Mah, S. Park, H. Kim, "Enhanced time-sync protocol for embedded sensor networks," *IEEE Vehicular Technology Conference*, 2006.
- [26] R. Urick, "Principles of underwater sound," *McGraw-Hill book company*, 1991.
- [27] I.F. Akyildiz, D. Pompili, T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc networks journal*, pp.257-279, March 2005.
- [28] G. Strang, "Linear algebra and its applications," *Brooks Cole*, 4th edition, July 19, 2005.
- [29] "Moteiv tmote sky quick start guide," <http://www.cems.uvm.edu/crobinso/mote/tmote-sky-quickstart-110.pdf>
- [30] "Trace collected from Tmotes," <http://anrg.usc.edu/wiki/>
- [31] V. Paxson, "On Calibrating Measurements of Packet Transit Times," *ACM SIGMETRICS*, Madison, Wisconsin, United States, p11 - 21, 1998.
- [32] K. Noh, Q. Chaudhari, E. Serpedin and Bruce W. Suter, "Novel Clock Phase Offset and Skew Estimation Using Two-Way Timing Message Exchanges for Wireless Sensor Networks," *IEEE TRANS. ON COMMUNICATIONS*, VOL. 55, NO. 4, APRIL 2007.
- [33] L. Lamport, "Time, clocks, and the ordering of event in a distributed system," *Communications of the ACM*, vol.21, Issue 7, p558 - 565, 1978.
- [34] L. Lamport and P. Melliar-Smith, "Synchronizing clocks in the presence of faults," *Journal of the ACM (JACM)*, vol.32, Issue 1, p52 - 78, 1985.
- [35] C. Ellingson and R. Kulpinski, "Dissemination of system time," *IEEE Trans. on Communications*, Vol. Com-21, No. 5, May 1973.
- [36] C. Ellingson and R. Kulpinski, "Dissemination of system time," *IEEE Trans. on Communications*, Vol. Com-21, No. 5, May 1973.
- [37] IEEE 1588 standard for time synchronization, "<http://ieee1588.nist.gov/>"



**Pai-Han Huang** obtained his M.S. in Electrical Engineering from University of Southern California in 2005, and is currently a Ph.D. student in the Department of Electrical Engineering-Systems at the University of Southern California. His ongoing research is focused on developing mathematical models for fine grained time synchronization protocols used in sensor networks, and energy optimal scheduling for wireless networks.



**Maulik Desai** obtained B.E. with Honors from The Cooper Union for the Advancement of Science and Art in 2007, and was awarded a full tuition scholarship for all four years. In the past he has participated in a team developing a digital receiver front-end for the MRI systems. His current research interest includes employing stochastic modeling techniques to enhance the performance of wireless networks. Maulik Desai is a member of Eta Kappa Nu, the honor society for the Electrical Engineers, and is currently a master student in the department of Electrical Engineering at the Columbia University.



**Xiaofan Qiu** attended Tsinghua University, Beijing, in 2003. After he finished sophomore year, he transferred to Texas A&M University, College Station, in 2005. Xiaofan Qiu is completing his senior year of undergraduate study, specializing in analogy circuit design. He is an Undergraduate Research Scholar at Texas A&M, and his scholar thesis is "Analysis and characterization of a programmable low-dropout regulator". He participated in a summer research at University of Southern California, in 2006, studying time synchronization protocols on wireless system. He is currently an engineer at Texas Instruments company.



**Bhaskar Krishnamachari** received the BE degree in electrical engineering from the Cooper Union, New York, in 1998 and the MS and PhD degrees from Cornell University in 1999 and 2002, respectively. He is currently an Associate Professor in the Department of Electrical Engineering, University of Southern California. His primary research interest is the design and analysis of efficient mechanisms for operating wireless networks.