Efficient Time Synchronization Mechanism for Wireless Multi Hop Networks

Bachar Wehbi, Anis Laouiti and Ana Cavalli
Institut TELECOM, TELECOM & Management SudParis, CNRS SAMOVAR
9 rue Charles Fourier, 91011 Evry Cedex, France
{bachar.wehbi, anis.laouiti, ana.cavalli}@it-sudparis.eu

Abstract— **This paper considers the time synchronization problem in wireless multihop networks. The objective is to design a protocol that synchronizes the network nodes with respect to one reference node. A set of sender nodes forming a connected dominating set is created to guide the synchronization process in a multihop environment. Each node estimates its clock offset and frequency error parameters to build an adjustment function that transforms its local time to that of the reference node. Our protocol estimates the clock offset and the frequency error on different time scales leading to significant gain in terms of synchronization accuracy. We give the theoretical bounds on the synchronization precision and show its performance through extensive simulations.**

Index Terms— **Time Synchronization, Computer Protocols, Multihop Networks, Wireless, Dominating Set, MPR, OLSR**

I. INTRODUCTION

Time synchronization is a crucial service for many applications and systems. The synchronization accuracy requirements differ depending on the applications in use. For example, CDMA2000 uses GPS to achieve the requirement for clock deviation not to exceed 1 part in 10^{10}. Internet application requirements are less strict. Network Time Protocol (NTP) [11] accuracy is in generally between $500\mu sec$ and $2msec$ [10]. This relatively high error is due to the non deterministic random time delay for a message transfer between Internet nodes and time servers.

In the recent years, new types of networks like wireless multihop sensor and adhoc networks have emerged. The needs for time synchronization in these networks differ depending on their usage. Sensor networks are tightly coupled to the physical world as their objective is to monitor real world phenomena like measuring temperature, humidity, pressure, etc. In adhoc networks, time synchronization is essential for applications like power saving, trace correlation for distributed monitoring, etc. In addition to these specific applications, these networks need time synchronization as typical distributed systems do for security protocols, coordination of tasks, ordering logged events, etc.

Many synchronization protocols have been proposed for wireless multihop networks [1], [2], [4]–[6], [8], [9], [13]–[15]. To improve the synchronization accuracy, most of existing proposals use MAC layer timestamping to reduce the sources of synchronization errors what makes them difficult to apply in networks where the radio interface is not bounded to the application layer. The IEEE 802.11 TSF [1] aims to maintain all stations with the same timer. In 802.11 adhoc mode, a node transmits a timestamped beacon if it wins the contention. A receiver of the beacon should adopt the contained TSF timer value if it is later than its own value. This leads faster nodes clocks drift to increase if they repeatedly loose beacon contentions. [8] and [14] improve TSF by giving faster nodes higher priority to send beacon frames. [13] does not synchronize the node clocks, instead, it generates and transforms timestamps using unsynchronized clocks to maintain a right chronology of events. Timing-sync [5] establishes a hierarchical structure in order to synchronize network nodes with respect to a reference node. A pair wise synchronization is performed along the edges of the structure. [9] does not create any structure to propagate the synchronization; when a node gets synchronized it starts synchronizing its non synchronized neighbor nodes. This makes it more robust to mobility however the synchronization overhead is high. [15] tries to exploit a constraint that has to be satisfied by a common notion of time in a multihop network. That is, for any loop of nodes the sum of offsets between two neighboring nodes along this loop must be equal to zero. By enforcing a large number of such constraints the synchronization accuracy can be improved. RBS [4] is a receiver to receiver mechanism that exploits the broadcast nature of wireless networks. Nodes send reference beacons to their neighbors using physical layer broadcasts. A reference broadcast does not contain an explicit timestamp; instead, receivers use its arrival time to compare their clocks. By considering the same message, RBS eliminates completely the non determinism caused by send and access time delays. Only the propagation and receive times affect the synchronization. The offset between any pair of nodes, receiving the beacon, is calculated by exchanging the local timestamps. Using RBS, every node stores information about the relative drift between its clock and the clock of any other node in the network. RBS is an interesting protocol for wireless multihop networks; however, it is mainly designed for one hop networks (though it gives an extension for multihop networks) and it requires each node to exchange time information with every node in its neighborhood.

Our objective is to design a protocol that achieves accurate and efficient network wide synchronization by exploiting the broadcast nature of the wireless medium. We design a protocol that extends the basic Reference Broadcast Synchronization (RBS) [4]. A structure of sender nodes responsible for organizing the synchronization process is constructed. Sender

978-1-4244-2644-7/08/$25.00 ©2008 IEEE

nodes are responsible for synchronizing their direct neighbors through the broadcast of reference messages. The synchronization process uses time information exchanged through reference messages to achieve in a short period of time an initial estimate of the clock offset and frequency error parameters. Then, by observing the offset estimate variation on longer time period, the mechanism improves the frequency error estimation leading to significant gain in synchronization precision. Using the estimated parameters each node builds an adjustment function that transforms its local clock value to that of the reference node. Our analysis gives us theoretical bounds on the achieved precision. We validate the mechanism with simulations showing very good performance in terms of accuracy and frequency error precision. Results show that the mechanism is robust against high packet loss what makes it a good candidate for synchronizing wireless multihop networks.

The rest of the paper is organized as follows. We present some basic concepts in section II. In section III we describe the proposed protocol. In section IV we analyze the clock offset and frequency error parameters estimation and we show that the estimation errors can be bounded. In section V we describe simulation results before we conclude in section VI.

II. CONCEPTS

A. Sources of Clock Imperfections

Crystal oscillators used in computer clocks are characterized by two important parameters: precision and stability. *Precision*, called frequency error, is the difference between the oscillator theoretical and real frequencies. it is given by the material manufacturer and is generally less than 10 parts in 10^6. *Stability* is the tendency of the oscillator to keep running over the same frequency over time. It is affected by environment factors like temperature variations, pressure, power supply voltage etc. On long term it is affected by the material aging. In general, time synchronization algorithms (e.g. NTP) try to model the oscillator as having a stable frequency error over a short period of observation and adjust it continuously to overcome the oscillator's instability.

B. Sources of Synchronization Error

Synchronization mechanisms use message exchange to estimate and adjust the clock parameters. The major source of synchronization error comes from the non deterministic time delay for a message transfer between two nodes. [7] decomposed the message latency into the following time delays: *Send Time* is the time spent to construct the message and transfer it to the network interface. *Access time* is the time spent by a packet waiting at the MAC layer for transmission. This delay is specific to the MAC protocol in use. *Propagation time* is the time taken by the wireless signals to traverse the medium from the sender to the receiver. *Receive time* is the time taken by the wireless interface to receive and notify the packet arrival. [4] shows that the difference between the receive times of a set of receivers follows a normal distribution $N(0, \sigma)$ where σ depends on the transmission baud rate.

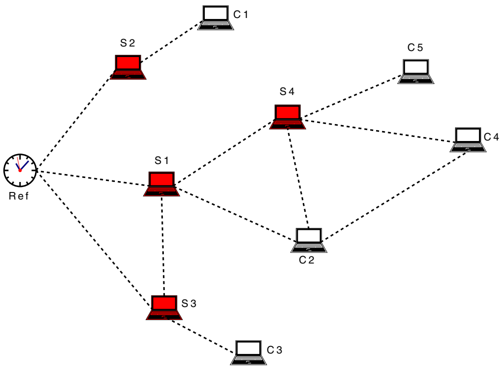


Fig. 1. A multi hop network example.

III. ACCURATE AND EFFICIENT SYNCHRONIZATION

In this section we describe the proposed time synchronization protocol. The objective is to support each network node with the required timing information in order to build an adjustment function that transforms its local clock value to that of the reference node. The reference node may be chosen based on administrative preferences or on an election mechanism. How this node is chosen is out of the scope of this paper; we assume it always exists. Using the adjustment functions they calculated, nodes, all over the network, run with similar clock values achieving therefore network wide synchronization. In order to reduce the sources of synchronization errors, we build our mechanism on RBS, a receiver to receiver mechanism, that by definition eliminates the *sender time* and *access time* errors. The mechanism consists of two complementary parts; the sender nodes selection and the synchronization process.

A. Sender Nodes Selection

The sender nodes selection consists of building a hierarchy of sender nodes in order to organize the synchronization process (a detailed description of the mechanism can be found in [16]). Sender nodes are responsible for transmitting reference messages. Receivers of a reference message use its arrival time to compare their clocks and estimate their relative offset. We construct a hierarchical structure of sender nodes rooted at the reference. A sender node at level 1 in the hierarchy (neighbor of the reference node) allows its clients at level 2 to get synchronized with respect to the reference node. For nodes more than two hops away from the reference, a sender node at level i allows its client nodes at level $i + 1$ to get synchronized with respect to its parent in the hierarchy which is a sender node at level $i-1$. This sender node playing now the role of time reference is called local reference node. To achieve network wide synchronization, each client must be neighbor of at least one sender node, therefore the structure of sender nodes should be a dominating set. Following this logic, the reference time can propagate hop by hop in the network achieving thus global synchronization with the reference node. Although any dominating set construction algorithm could be used, in this paper, we exploit the notion of Multi Point Relay (MPR) [12] used in OLSR [3] to build the sender nodes structure. Sender nodes at level 1 are the MPR nodes of the reference node. Sender nodes at level 2 are the MPR nodes of the sender nodes at level 1. Similarly sender nodes at level i are the MPR nodes of sender nodes at level $i-1$. Figure 1 illustrates the roles of the nodes in the synchronization process. Nodes S_1, S_2 and S_3 are sender nodes of level 1; they will synchronize their neighbors with respect to the reference node Ref. Where, S_4 is sender node of level 2; it will synchronize its neighbors C_4 and C_5 with respect to its local reference node S_1, its parent in the sender nodes hierarchy.

B. Synchronization Process

Sender nodes initiate the synchronization process and repeat it periodically. Each sender node broadcasts reference messages at predefined time interval. Upon reception of a reference message, the local reference and the client nodes record the message's arrival time. Then, the local reference node sends a message containing the recorded reception time, T_{LR}, to the sender node who rebroadcasts it to its client nodes. Using this information, each client node constructs a $TimeTable$ that contains for each received reference message, M_i, the mapping between its local reception time $T_{C,i}$ and that of the local reference node $T_{LR,i}$. When the $TimeTable$ holds a specified number of entries, the node uses the least squares linear regression to estimate the best fit line relating the node's clock to the local reference node's clock. The estimated best fit line is an adjustment function that transforms the client's local clock value to that of the local reference node and therefore to that of the reference node. This adjustment function is given by equation 1 below:

$$T_{synch} = (1 + \widetilde{F}) \times T_{local} + \widetilde{Off} \tag{1}$$

Where $\widetilde{F}$ and $\widetilde{Off}$ are the estimated frequency error and offset parameters respectively. This process, that we call $Cycle$, of recording in the $TimeTable$ the mapping between local reception times and local reference node reception times of reference messages is repeated a predefined number of times. At the end of each $Cycle$, the node re-estimates its Offset parameter, adjusts accordingly the adjustment function and records in a $CycleTable$ the estimated offset value, $\widetilde{Off}$, and the cumulated offset value, $COff$.

$$Cycle_i : \widetilde{Off}_i, COff_i$$

Where $COff_i = COff_{i-1} + \widetilde{Off}_i$ and $COff_1 = \widetilde{Off}_1$. This offset re-estimation process is used to improve the frequency error estimation precision. The $CycleTable$ table contains the offset variation[1] on longer time period than the $TimeTable$. Considering longer term observations can considerably improve the frequency error estimation precision as we will see in section IV-C. Once the $CycleTable$ contains a specified number of entries, the node applies linear regression to re-estimate the frequency error, $\widetilde{F}$, and adjusts the adjustment function accordingly. To resume, the synchronization process uses time information exchanged through reference messages to achieve in a short time period an initial estimate of the node's adjustment function. Then, by observing the offset estimate variation on longer time period, it improves the frequency error estimation and therefore the time synchronization accuracy as we will see in the next sections.

Using the described synchronization process, client nodes do not need to transmit any control messages as opposed to all sender to receiver synchronization where each client exchanges control messages with the reference, and to RBS where client nodes exchange between themselves time information. This reduces highly the message exchange.

IV. ANALYSIS

In this section we analyze the Offset (section IV-A) and the frequency error (section IV-B) estimations and we show how they are related and we give theoretical bounds (section IV-C) on the precision of these estimates.

A. Offset Estimation

As described in section III-B, each client node maintains a table that, for every received reference message M_i, maps the local reception time $T_{C,i}$ to the local reference node reception time $T_{LR,i}$. Assuming the propagation error is negligible, equation 2 gives, for a given reference message, the measured offset O_i between the client's and reference's reception times.

$$O_i = T_{C,i} - T_{LR,i} = Off_{T_{C,i}} + \epsilon_r \tag{2}$$

Where $Off_{T_{C,i}}$ is the real offset at the reception of the reference message and ϵ_r is the receive error which follows a normal distribution [4]. To indicate that the offset depends on the frequency error F, equation (2) can be written as:

$$Off_{T_{C,i}} = Off_{T_{C,j}} + F * (T_{C,i} - T_{C,j}) \tag{3}$$

In order to estimate the clock offset, an estimation $\widetilde{F}$ of the frequency error is required. Let's consider we have this estimate; equation 4 shows the clock offset estimation, $\widetilde{Off}$, at the reception of the n_{th} reference message (end of a $Cycle$) in function of the frequency error estimate.

$$\widetilde{Off} = \bar{O} + \widetilde{F}(T_{C,n} - \bar{T_C}) \tag{4}$$

Equation 4 can be written as follows:

$$\widetilde{Off} = \frac{\sum_{i=1}^{n} O_i}{n} + \widetilde{F}(T_{C,n} - \frac{\sum_{i=1}^{n} T_{C,i}}{n})$$

$$\widetilde{Off} = \frac{\sum_{i=1}^{n} (Off_{T_{C,i}} + \epsilon_r) + \sum_{i=1}^{n} \widetilde{F}(T_{C,n} - T_{C,i})}{n}$$

$$\widetilde{Off} = \frac{\sum_{i=1}^{n} (Off_{T_{C,n}} + F(T_{C,i} - T_{C,n}) + \epsilon_r) + \sum_{i=1}^{n} \widetilde{F}(T_{C,n} - T_{C,i})}{n}$$

$$\widetilde{Off} = Off_{T_{C,n}} + \Delta_F(T_{C,n} - \bar{T_C}) + \epsilon_{r,n} \tag{5}$$

Equation 5 shows that the estimated offset is affected by the reception error (average over n values), the precision of the frequency error estimation and the duration of the synchronization mechanism. In the next section we will detail the estimation of the frequency error.

---

[1] The offset variation is the impact of frequency error; the variation of the first can be used to estimate the second.

B. Frequency Error Estimation

The clock adjustment processing consists of estimating the clock frequency error and the relative offset. Section IV-A showed the offset estimation depends on that of the frequency error where the need for an accurate frequency error estimation. The clock's frequency may change over time due to environmental factors (see section II-A). This makes it harder to model the oscillator's frequency change. To simplify the problem, it is widely modeled as of having a high stability on short term. In other words, the oscillator's frequency is assumed not to change over a short period of time. On longer term, repeating the synchronization mechanism ensures the frequency error is continuously re-estimated. In the following we assume the clock frequency is stable over the period of observation.

The frequency error is estimated using the best fit line least squares linear regression. This is a statistical tool where the objective is to plot the best fit line that relates the client node's reception times $T_{C,i}$ to the correspondent offset values O_i of equation 2. Equation 6 presents the details of the frequency error estimation $\widetilde{F}$:

$$\widetilde{F} = \frac{\sum_{i=1}^{n}(T_{C,i} - \bar{T_C})(O_i - \bar{O})/n}{\sum_{i=1}^{n}(T_{C,i} - \bar{T_C})^2/n} \tag{6}$$

The precision of the frequency error estimation depends on three parameters: the duration of the synchronization mechanism reflected by the sample of $T_{C,i}$ values, the number of time points reflected by the value n and the reception error included in the measured offset values O_i as expressed in equation 2.

The performance of the synchronization mechanism depends highly on the precision of the frequency error estimation. This precision depends on the understanding of the effects of these three parameters on the frequency error; this is what we will analyze in the next section.

C. Mathematical analysis

In this section we show that the estimation error of the frequency error and offset estimations can be theoretically bounded. Using equations 2 and 3 give:

$$(O_i - \overline{O}) = F \cdot (T_{C,i} - \overline{T_C}) + \epsilon_r - \epsilon_{r,n}$$

Using the previous equality in equation 6 gives:

$$\Rightarrow \widetilde{F} = \frac{\sum_{i=1}^{n}\{(T_{C,i} - \overline{T_C}) \cdot [F \cdot (T_{C,i} - \overline{T_C}) + \epsilon_r - \epsilon_{r,n})]\}}{\sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2}$$

$$\widetilde{F} = \frac{F \cdot \sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2 + \sum_{i=1}^{n}(T_{C,i} - \overline{T_C}) \cdot (\epsilon_r - \epsilon_{r,n})}{\sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2}$$

$$\Rightarrow \Delta F = \widetilde{F} - F = \frac{\sum_{i=1}^{n}(T_{C,i} - \overline{T_C}) \cdot (\epsilon_r - \epsilon_{r,n})}{\sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2} \tag{7}$$

If we suppose that the reference message exchange is periodic of period P we get that:

$$\sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2 = \left\{ \begin{array}{ll} 2 \cdot \sum_{i=1}^{(n-1)/2}(P \cdot i)^2 & n\ odd \\ 2 \cdot \sum_{i=0}^{(n-2)/2}(P \cdot (1+2i)/2)^2 & n\ even \end{array} \right\}$$

Developing the previous expression we get:

$$\sum_{i=1}^{n}(T_{C,i} - \overline{T_C})^2 = \frac{n \cdot (n^2-1)}{12} \cdot P^2 \qquad \forall n \tag{8}$$

On the other hand, we know that the sum of two independent normal distributions, $N(0, \sigma_1)$ and $N(0, \sigma_2)$, is a normal distribution $N(0, \sqrt{\sigma_1^2 + \sigma_2^2})$. Accordingly, the numerator of the expression ΔF (equation 7) can be written as (details can be found in [16]):

$$\sum_{i=1}^{n}(T_{C,i} - \overline{T_C}) \cdot (\epsilon_{r,i} - \epsilon_{r,n}) = N(0, \sigma \cdot P \cdot \sqrt{\frac{n \cdot (n^2-1)}{12}}) \tag{9}$$

(8) and (9) give:

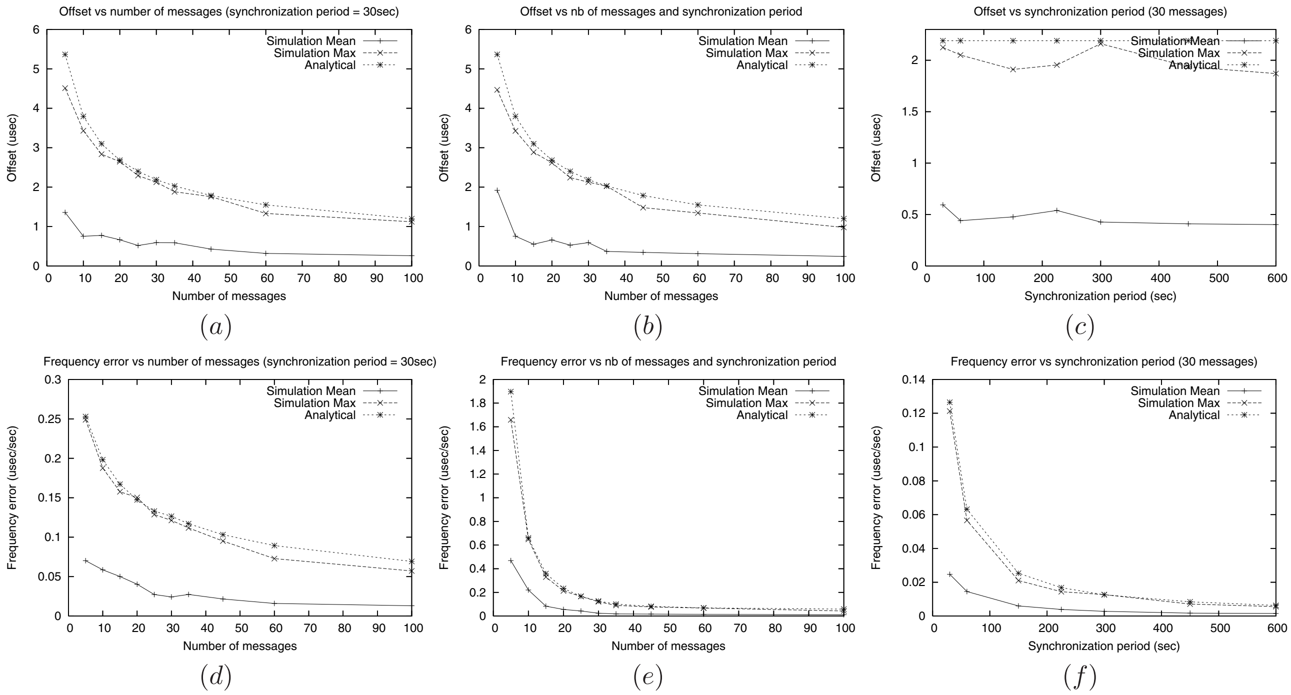
$$\Delta F = \widetilde{F} - F = N(0, \frac{2\sqrt{3} \cdot \sigma}{\sqrt{n \cdot (n^2-1)} \cdot P}) \tag{10}$$

Accordingly, the estimation error of the frequency error estimation can be probabilistically bounded. Equation (10) and equation (5) give:

$$\Delta Off = \widetilde{Off} - Off_{T_{C,n}} = N(0, \frac{2 \cdot \sigma}{\sqrt{n}})$$

Similarly, the estimation error of the offset estimation can be probabilistically bounded. After this mathematical analysis we performed a set of simulations in order to study the variation of the frequency error and offset estimations with the number of exchanged reference messages and the synchronization period. Figure 2 shows the results given by the theoretical bounds (fidelity of 99%), the simulations maximum and mean values. Figure 2-c shows, as expected, that the Offset estimation does not depend on the synchronization period; its precision increases with the number of messages, this increase starts to diminish for $n > 30$ (figures 2-a and 2-b). On the other hand the frequency error precision grows faster with n; it reaches a clear point of diminishing return for $n > 30$ (figure 2-e). When we fix the synchronization period (product $n.P$) and vary n and P accordingly, the frequency error precision increases less clearly than in figure 2-e but similarly the increase diminishes slightly for n higher than 30. The interesting result is illustrated in 2-f; the frequency error precision increases with the synchronization period $n.P$ for a fixed number of messages n this increase diminishes for $n.P$ higher than 300 seconds.

The aim of the mechanism is to increase the synchronization accuracy; that is the accuracy of the offset and the frequency error estimates. This can be achieved by increasing the number of exchanged reference messages n or increasing the synchronization period P. However, increasing n means higher bandwidth consumption which is a limited resource in a wireless network. While increasing the synchronization period means that a node has to wait more time in order to build its adjustment function required for the synchronization. Based on the results shown in figure 2 and precisely on the fact that the offset estimate does not depend on P and from the fact that the frequency error estimate is improved when considering longer observations, we design our mechanism to separate these two estimates. A non synchronized node needs



(a) (b) (c)

(d) (e) (f)

Fig. 2. Variation of the Offset and Frequency error estimations when we vary the number of reference messages n, the reference message transmission rate P and the synchronization period $n \cdot P$.

in a short period of time to build its adjustment function, thus, based on short period observation (based on information gotten through reference message exchange) a node estimates its clock offset and its frequency error. Then, on observing the variation of the offset estimate on longer time period, it fine tunes the frequency error estimate. This procedure allows a node to build an initial estimate of it's adjustment function in a short period of time (i.e. 30 seconds) and fine tune it when enough observations are available (i.e. 300 seconds).

V. PROTOCOL EVALUATION

To evaluate the performance of our synchronization protocol, we have developed a simulator that implements the synchronization process and the sender nodes selection described earlier. 250 nodes are randomly placed on a 2x2units square and each node has a transmission range of 0.25 units. The reference node is placed at the center of the square. At initialization, each node has an offset chosen randomly in the range of [-5min, +5min] and a frequency error chosen randomly in the range of [-10 $\mu sec/sec$, + $10\mu sec/sec$]. The wireless medium is simulated as having a random and independent packet loss ratio. Sender nodes transmit references messages periodically (1 message per second) using 1Mb/s bit rate. For this bit rate the receiver error standard deviation σ is approximately equal to $2\mu sec$. Client nodes collect time information by observing reference messages for $30 seconds$, then they estimate their clock offset and frequency error (if no estimation exists). Then they re-estimate their offset by considering always 30 seconds observation window. After 10 offset estimates (300 seconds) they re-estimate their frequency error and start over the mechanism for 3000 seconds; at this time all the nodes stop the synchronization mechanism.

	Our mechanism			RBS synchronization		
Nb hops	Mean	Stdv	Max	Mean	Stdv	Max
1 hop	0.6	0.75	2.58	1	1.31	6.41
2 hops	0.76	0.95	3.50	1.91	2.27	14.47
3 hops	0.87	1.1	4.41	2.74	3.49	16.30
4 hops	0.97	1.21	4.80	3.43	4.85	20.15
5 hops	1.03	1.30	5.39	4.08	5.63	26.33

TABLE I

PRECISION IN μsec OF THE OFFSET ESTIMATION FOR OUR SYNCHRONIZATION MECHANISM AND RBS.

A. Synchronization Accuracy

In this section we compare the synchronization accuracy of our mechanism to RBS. Table I shows the mean, standard deviation and the maximum value of the offset estimation in μsec when using our mechanism and RBS for nodes at different hops away from the reference node. The results show that the offset precision of our mechanism outperforms RBS and it is less sensitive to the number of hops. This is because our mechanism provides better estimate of the frequency error than RBS, and as we have shown previously the offset estimation depends on that of the frequency error.

Table II shows that our mechanism provides a better estimate of the frequency error than RBS; in global the frequency error estimate precision is $36 times$ higher than RBS. This is because our mechanism improves the frequency error estimate by considering observations on longer time period.

To interpret these results, assume the application in use requires that synchronization error does not exceed $100\mu sec$. While the synchronization process is running, our mechanism has a maximum offset error of $5.39\mu sec$ and a maximum frequency error estimation error of $4.710^{-3}\mu sec/sec$. Ac-

	Our mechanism		RBS synchronization	
Nb hops	Mean	Max	Mean	Max
1 hop	$6.624\ 10^{-4}$	0.0016	0.0252	0.0629
2 hops	$9.977\ 10^{-4}$	0.0026	0.0457	0.1089
3 hops	$1.2\ 10^{-3}$	0.0032	0.0480	0.1232
4 hops	$1.4\ 10^{-3}$	0.0042	0.0458	0.1575
5 hops	$1.6\ 10^{-3}$	0.0047	0.0552	0.2362

TABLE II

PRECISION IN PARTS PER MILLION OF THE FREQUENCY ERROR ESTIMATION FOR OUR MECHANISM AND RBS.

	Our mechanism				RBS synchronization			
Nb hops	0%	10%	20%	50%	0%	10%	20%	50%
1 hop	0.6	0.63	0.7	1.01	1	1.14	1.27	2.12
2 hops	0.76	0.84	0.94	1.39	1.91	2.12	2.54	4.35
3 hops	0.87	0.97	1.1	1.68	2.74	3.06	3.62	6.89
4 hops	0.97	1.06	1.22	1.94	3.43	3.88	4.55	9.91
5 hops	1.03	1.15	1.35	2.19	4.08	4.70	5.45	13.35

TABLE III

EFFECT OF PACKET LOSS ON THE MEAN PRECISION μSEC OF THE OFFSET ESTIMATION FOR OUR MECHANISM AND RBS.

Our synchronization mechanism				
Nb hops	No loss	10% loss	20% loss	50% loss
1 hop	$6.6\ 10^{-4}$	$9.7\ 10^{-4}$	$1.2\ 10^{-3}$	$2.3\ 10^{-3}$
2 hops	$9.9\ 10^{-4}$	$1.2\ 10^{-3}$	$1.4\ 10^{-3}$	$2.8\ 10^{-3}$
3 hops	$1.2\ 10^{-3}$	$1.4\ 10^{-3}$	$1.7\ 10^{-3}$	$2.6\ 10^{-3}$
4 hops	$1.4\ 10^{-3}$	$1.7\ 10^{-3}$	$2.0\ 10^{-3}$	$3.7\ 10^{-3}$
5 hops	$1.6\ 10^{-3}$	$2.1\ 10^{-3}$	$2.2\ 10^{-3}$	$4.9\ 10^{-3}$
RBS synchronization mechanism				
1 hop	0.0252	0.0377	0.0500	0.0854
2 hops	0.0457	0.0493	0.0613	0.0872
3 hops	0.0480	0.0512	0.0692	0.1082
4 hops	0.0458	0.0593	0.0793	0.1367
5 hops	0.0552	0.0669	0.1030	0.2118

TABLE IV

EFFECT OF PACKET LOSS ON THE MEAN PRECISION (IN PARTS PER MILLION) OF THE FREQUENCY ERROR ESTIMATION FOR OUR MECHANISM AND RBS.

cordingly, once we stop the synchronization mechanism, the network can respect the application requirement for a period of 20129 sec (5 h 35 min 29 sec). On the other hand, RBS has a maximum offset error of $26\mu sec$ and a maximum frequency error estimation error of $2.310^{-1}\mu sec/sec$, Thus, using RBS, the network can respect the application requirement for $321sec$ which is by far less than what our mechanism allows.

B. *Effect of Packet Loss*

In this section, we study effect of packet loss on our mechanism and RBS. Table III shows the effect of different packet loss rates on the offset precision of our mechanism and RBS. We can see that our protocol is more robust to packet lost than RBS especially in lousy channel condition (high packet loss). We should note here that the absolute value of the offset precision for our mechanism is clearly higher than RBS. Facing 50% packet loss, our mechanism has an offset error comparable to that of RBS in perfect channel conditions. Table IV shows the variation of the precision of the frequency error estimation with the packet loss ratio for RBS and our mechanism. Again, the absolute value of the frequency error precision of our mechanism facing 50% packet loss is by far higher than RBS in perfect channel conditions.

VI. CONCLUSION

We have presented a time synchronization protocol for wireless multihop networks. The proposed mechanism allows each node to build an adjustment function that transforms its time to that of the reference node. This mechanism reduces the communication overhead by limiting the number of nodes that transmit control messages. The mechanism works on two time scales; in a short time period it provides initial estimates of the frequency error and offset parameters while observations on longer period allow to improve significantly the parameters accuracy. We proved through mathematical analysis that the precision of the synchronization can be theoretically bounded. Simulation results show significant improvements of the synchronization accuracy over the basic RBS mechanism. The maximum synchronization error is in the order of few μsec even for nodes multiple hops away from the reference node. Even in a lousy channel situation, our mechanism still keeps the network accurately synchronized. Our future work consists on studying the effect of mobility and the change of the reference node on the synchronization.

REFERENCES

[1] IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007.

[2] H. Attiya, D. Hay, and J. L. Welch. Optimal Clock Synchronization Under Energy Constraints in Wireless Ad-Hoc Networks. In *OPODIS*, pages 221–234, 2005.

[3] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, Oct. 2003.

[4] J. Elson, L. Girod, and D. Estrin. Fine-Grained Network Time Synchronization Using Reference Broadcasts. In *OSDI*, 2002.

[5] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync Protocol for Sensor Networks. In *SenSys*, pages 138–149, 2003.

[6] H. Kim, D. Kim, and S. Yoo. Cluster-based Hierarchical Time Synchronization for Multi-hop Wireless Sensor Networks. In *AINA (2)*, pages 318–322, 2006.

[7] H. Kopetz and W. Schwabl. Global Time in Distributed Real-time Systems. Technical Report 15/89, Technische Universitat Wien, 1989.

[8] T.-H. Lai and D. Zhou. Efficient and Scalable IEEE 802.11 Ad-Hoc-Mode Timing Synchronization Function. In *AINA*, pages 318–323, 2003.

[9] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The Flooding Time Synchronization Protocol. In *SenSys*, pages 39–49, 2004.

[10] D. L. Mills. Precision Synchronization of Computer Network Clocks. *SIGCOMM Comput. Commun. Rev.*, 24(2):28–43, 1994.

[11] D. L. Mills. *Computer Network Time Synchronization: the Network Time Protocol*. CRC Press, 2006.

[12] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. In *HICSS*, page 298, 2002.

[13] K. Römer. Time Synchronization in Ad Hoc Networks. In *MobiHoc*, pages 173–182, 2001.

[14] J.-P. Sheu, C.-M. Chao, and C.-W. Sun. A Clock Synchronization Algorithm for Multi-Hop Wireless Ad Hoc Networks. In *ICDCS*, pages 574–581, 2004.

[15] R. Solis, V. S. Borkar, and P. R. Kumar. A New Distributed Time Synchronization Protocol for Multihop Wireless Networks. In *CDC*, pages 2734–2739, 2006.

[16] B. Wehbi, A. Laouiti, and A. Cavalli. Accurate and efficient time synchronization mechanism for wireless multi hop networks. Technical report, TELECOM SudParis, 2008.