Novel Method of Time Synchronization based on IEEE 1588

1Xiaoyan Zhuang, 2Zhijian Dai, 3Houjun Wang
1School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
xiaoyzhuang@gmail.com
2,3School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
doi:10.4156/jdcta.vol5.issue7.34

Abstract
Time synchronization based on IEEE 1588 synchronization protocol aims to synchronize the slave clock to the master clock. Conventional synchronization algorithm focuses on calibrating offset of the slave clock with respect to the master clock. In this paper, a novel method of time synchronization is proposed, the proposed method improves the synchronization precision by considering both offset calibration and drift compensation. Asymmetric communication between downlink and uplink data rate is employed in time synchronization. Experimental results indicate that our proposed method can reduce the increase speed of offset between the slave clock and the master clock, and it outperforms the conventional synchronization method.

Keywords: IEEE 1588, Time Synchronization, Drift Clock Model, Asymmetric Communication

1. Introduction
IEEE 1588 is a standard to synchronize independent clocks running on separate nodes of a distributed wireless network and control system [1, 2]. IEEE 1588 can be employed as key Synchronization technology in Packet Transport Network (PTN). IEEE 1588 is a master-slave synchronization protocol. After a master-slave hierarchy has been established, time stamped messages are exchanged between the master nodes and slave nodes to enable the slave node to measure the time on master clock [3, 4]. Accordingly, all network nodes can synchronize to the reference clock.

In time synchronization based on IEEE1588, most current works [5-7] focus on calibrating offset between master clock and slave clock. While in real world case, time synchronization precision is prone to drift of frequency of crystal oscillator beside its offset. M. D. Lemmon [8] employed the drifting clock model in time synchronization, which considered both offset and drift achieving higher precision than that of without considering drift of clock. In time synchronization, slave clock is synchronized to master clock by massages exchange between master node and slave node. The communication links of massages exchange are assumed to be symmetric in IEEE 1588 time synchronization protocol, which means that the downlink and uplink are with the same propagation delay. However, due to the impact of environment, the propagation delay of downlink and uplink data rate is different, and asymmetric communication links would be more accurate to characterize the behavior of packet transport between master node and slave node. Shuai Lv [9] calibrated offset based on asymmetric communication link without considering the drift of clock.

In this paper, we present a novel method to synchronize the slave clock based on IEEE 1588 synchronization protocol. Besides offset calibration, we employ drift compensation in time synchronization, which updates the slave clock model to reduce the increase speed of the difference between the slave clock and the master clock. Drift compensation is implemented by the messages exchanged in offset calibration, and it does not add communication cost. The proposed method calibrates the offset and compensates the drift of slave clock by exchange of massages with asymmetric communication links.

The rest of paper is organized as follows: Clock model is described in Section II. In Section III, we review offset calibration and propose drift compensation, and the implementation of drift compensation is discussed. Experimental simulations are reported in Section IV. And the conclusion is given in Section V.
2. Clock Model

In time synchronization, slave clock is synchronized to master clock based on a deterministic clock model. In practice, clock model can be characterized using two parameters [10]: skew and offset. Skew refers to the rate at which clock measures time, and offset refers to the difference between the local times of two clocks that have the same skew. The time model of clock is represented as follow.

\[ C(t) = at + b \]  

where \( a \) and \( b \) are skew and offset respectively. The time on master node \( C_m(t) \) and slave node \( C_s(t) \) can be derived from (1), whose representation can be expressed as follow:

\[ C_m(t) = a_1t + b_1, \]  

and

\[ C_s(t) = a_2t + b_2. \]

According to (2) and (3), we can establish the relations between the time on master node \( C_m(t) \) and slave node \( C_s(t) \), and it can be written as:

\[ C_s(t) = \frac{a_2}{a_1} C_m(t) + (b_2 - \frac{a_2}{a_1}b_1). \]  

Since the purpose of time synchronization is to synchronize the slave clock to the master clock, the time on master clock can be referred as the reference clock or the standard clock. For simplifying the expression, we revise the time model on master node as follow:

\[ C_m(t') = t' \]  

Time synchronization aims to synchronize the time on slave clock to the time on master clock, which means that we calibrate the time on slave clock \( C_s(t) \) to the reference time \( t' \) by using the synchronization algorithms. Substituting (5) into (4), we have

\[ C_s(t') = \frac{a_2}{a_1} t' + (b_2 - \frac{a_2}{a_1}b_1). \]  

\( a_2/a_1 \) and \( b_2-(a_2/a_1)b_1 \) can be treated as skew and offset of slave clock respectively, and the time on slave clock is a function of the reference time. In time synchronization, in order to synchronize the time on slave clock to master clock, we need to calibrate skew to “1” and offset to “0”. Generally, in clock modeling, skew consists of two parts: a constant “1” and a variable drift \( \varepsilon \). We revise (6) as follow:

\[ C_s(t') = (1 + \varepsilon)t' + b \]  

where \( \varepsilon \) denotes drift of slave clock relative to the master clock, and \( b \) is the initial offset between slave clock and master clock at the reference time \( t' = 0 \). The difference between the time on slave clock and the time on master clock is a criterion to evaluate the performance of time synchronization, which can be defined as:

\[ Offset = C_m(t') - C_s(t') \]

\[ = \varepsilon \cdot t' + b \]  

Time synchronization minimizes the difference between the time on slave clock and the time on master clock. The difference is calibrated as offset of slave clock. In (8), we noticed that,
due to the existence of drift, the difference increases with the reference time $t'$. In time synchronization, if we only consider calibrating offset, though the time on slave clock could achieve high precision at the moment of calibration, the increase speed of the difference is not reduced. Drift is a variable that is prone to the uncertainty of environment. However, the duration between each calibration is very short (generally 2 seconds), drift can be assumed to be a constant in the short period of time. We can compensate drift in time synchronization to minimize the difference.

3. Time Synchronization

3.1. Offset Calibration

As analyzed in Section II, offset is a function of drift $\varepsilon$, we need to calibrate offset with a deterministic interval. There is a trade-off between the interval and the synchronization precision: if the interval is very short, higher synchronization precision can be achieved and more serious communication burden would bring to network. In time synchronization based on IEEE 1588, offset is calibrated in every 2 seconds. The time synchronization is implemented by the two-way messages exchange, and slave node derives the reference time from master node to compute and calibrate offset. A basic block scheme of time synchronization is depicted in Figure 1.

![Figure 1. The basic model of Time Synchronization](image)

In each calibration, master node sends a synchronization ($Sync$) message to slave node at a fixed period of 2 seconds. Master clock measures the time $T_{ms1}$ at the moment which the $Sync$ message is sent out. Once slave node receives the $Sync$ message, slave clock measures and stores the receiving time $T_{sr1}$. Master node sends a $Follow Up (T_{m1})$ message that contains the accurate time $T_{ms1}$. Slave node sends a $Delay_Req$ message and stores the time $T_{ss2}$ at the moment which the $Delay_Req$ message is sent out. When master node receives the $Delay_Req$ message, it measures the receiving time $T_{mr2}$ and sends a $Delay_Resp (T_{m2})$ message back to slave node, and the $Delay_Resp$ message contains the time stamp $T_{mr2}$. Since there are messages from master node to slave node and from slave node to master node, the time synchronization is called two-way exchange.
Based on the above analysis, the synchronization process of Precise Time Protocol (PTP) mainly consists of two phases: deriving the clock offset from master node and computing the propagation delay between master node and slave node.

In phase 1, we derive the difference. Due to the existence of the data message submit retard in the networks, there is duration between message packaging and message sending, and we cannot stamp the accurate sending time \(T_{ms1}\) in the Sync message. The Follow Up message is employed to meet this challenge, in which \(T_{ms1}\) is stamped. Slave clock measures receiving time \(T_{sr1}\) when the Sync message reaches slave node. In practice, the propagation delay cannot be ignored, which is a part of the difference. The difference can be expressed as

\[
T_{sr1} - T_{ms1} = Offset + D_{ms}
\]  

where \(Offset\) is the offset between slave clock and master clock, \(D_{ms}\) is the propagation delay from master node to slave node.

### 3.2. Drift Compensation

As analyzed in clock model, the difference between the time on slave clock and the time on master clock can be expressed as a function of drift \(\varepsilon\). Though offset calibration, we can synchronize the time on slave clock to the time on master clock. However, offset calibration does not reduce the increase speed of the difference. Since the drift \(\varepsilon\) can be assumed to be a constant in the short period of time, we can update the slave clock model to reduce the increase speed of the difference by employing drift compensation.

![Figure 2. The process of drift compensation](image)
In this subsection, we consider drift compensation in time synchronization. A possible block scheme of drift compensation is shown in Figure 2. Drift compensation is implemented through messages propagation from master node to slave node and adds no communication cost. In drift compensation, we only need the Sync and Follow Up messages, which are transmitted from master node to slave node in offset calibration. When the first Sync message is received at $T_{sr1}$ in slave node, we calibrate offset by (12) and update the slave clock model (7) as

$$C_s(t) = (1 + \varepsilon_1)C_m(t) + \text{Offset}_1,$$

and the calibrated receiving time $T_{sr1}'$ can be written as

$$T_{sr1}' = (1 + \varepsilon_1)T_{m1} + D_m + \text{Offset}_1,$$

where $\text{Offset}_1$ is the initial difference after the first offset calibration, and $D_m$ is the propagation delay from master node to slave node.

In the first calibration, we do not compensate drift. The second Sync message is received by slave node at $T_{sr2}$, from (5) and (13), $T_{sr2}$ can be expressed as

$$T_{sr2} = (1 + \varepsilon_1)T_{m2} + D_m + \text{Offset}_1,$$

We first calibrate offset, and drift is compensated as

$$\varepsilon_2 = \frac{T_{sr2} - T_{sr1}'}{T_{m2} - T_{m1}} - 1,$$

and slave clock is updated as

$$C_s(t) = (1 + \varepsilon_2)C_m(t) + \text{Offset}_2,$$

and obtain the updated receiving time $T_{sr2}'$.

After the $i$-th Sync message is received by slave node at $T_{sr(i)}$, drift can be updated as

$$\varepsilon_i = \frac{T_{sr(i)} - T_{sr(i-1)\prime}}{T_{m(i)} - T_{m(i-1)}} - 1,$$

and slave clock model can be revised as follow

$$C_s(t) = (1 + \varepsilon_i)C_m(t) + \text{Offset}_i.$$

### 3.3. Asymmetric Communication Links

In offset calibration, the communication links between slave node and master node are assumed to be symmetric, and the propagation delay $D_{mr}$ equals to $D_{rm}$. While in the practice application, the propagation delay from master node to slave node is different from the propagation delay from slave node to master node, and the communication links are asymmetric.
In this paper, asymmetric ratio is adopted to characterize the communication links, which is introduced by Lee. Figure 3. depicts asymmetric message propagation, and it is intensively discussed in [9]. The asymmetric ratio \( R \) can be calculated as follow:

\[
R = \frac{T_{m4} - T_{m3}}{T_{s4} - T_{s3}} \quad (20)
\]

Offset can be derive from (12) and (20),

\[
Offset = T_{s1} - T_{m1} - \frac{D_{m} + D_{s}}{1 + R} = R(T_{m1} - T_{s1}) - (T_{m2} - T_{s2}) \quad (21)
\]

In drift compensation, there is only message transmission from master node to slave node. We do not need to consider the asymmetric communication links.

4. Experimental Results

Experiments were reported to evaluate the performance of the proposed method. In all experiments, the communication link speed and the initial offset were set to 100 ms and 60 s respectively. The frequency of slave clock was 25 MHz, whose clock skew was 40 ppm, and the frequency of master clock was 50 MHz. The Sync interval was set to 2 s. In order to compare the accuracy of the proposed approach and the conventional synchronization algorithms, bias error of the time on slave clock with respect to master clock is introduced,

\[
BiasError = |IdealTime - EstimatedTime| \quad (22)
\]

We first considered drift compensation in time synchronization. In this experiment, the communication links were assumed to be symmetric. Since the purpose of drift compensation is to reduce the increase speed of the difference between the time on slave clock and the time on master.
clock, the $BiasError$ is calculated before offset calibration, which is the maximal value of bias error in each synchronization period. Figure 4(a). depicts the bias error without drift compensation, the average and maximal $BiasError$ are $9.32 \times 10^{-10}$ s and $1.07 \times 10^{-9}$ s respectively. Figure 4(b). depicts the bias error with drift compensation, the average and maximal $BiasError$ are $1.45 \times 10^{-11}$ s and $1.68 \times 10^{-11}$ s respectively.

In the second experiment, we considered both the asymmetric communication links and drift compensation in time synchronization. The average bias error is introduced to evaluate the performance of the proposed method:

$$Average\ Bias Error = \frac{|BiasError|_b + |BiasError|_a}{2},$$  \hspace{1cm} (23)

where $BiasError|_b$ and $BiasError|_a$ are $BiasError$ calculated before offset calibration and after offset calibration respectively. In this experiment, the asymmetric ratio was set to different values (from 1:1 to 8:1). Figure 5. shows the average bias error of the time on slave clock as a function of the asymmetric ratio. The experimental result indicates that the proposed method is significantly outperforms the convention algorithm.
5. Conclusion

Time synchronization aims to synchronize the slave clock to the master clock. The conventional algorithms only consider offset calibration. In this work, we investigate a method by employing drift compensation. The proposed method updates the slave clock model in time synchronization, which is implemented by messages exchange. All of these messages are transmitted in offset calibration, and it does not add any communication cost. With the help of drift compensation, the increase speed of the difference between the time on slave clock and the time on master clock is reduced. In practical application, the communication links are asymmetric. Though combining drift compensation and asymmetric communication, the precision of time synchronization is significantly improved.

6. Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant 60827001.

7. References