OMNeT++ Based Modeling and Simulation of the IEEE 1588 PTP Clock

Yingshu Liu
School of Electrical Engineering and Automation
Tianjin University
Tianjin, China
liu_ysh@tju.edu.cn

Cheng Yang
School of Electrical Engineering and Automation
Tianjin University
Tianjin, China
yangcheng19860201@163.com

***Abstract*—Time synchronization has played an important role in modern measurement and control systems which require high time accuracy to maintain optimal performance in sample acquisition, signal control and safety protection. The Ethernet/IP based IEEE 1588 precise time protocol (PTP) provides us a prospective solution. By means of taking hardware timestamps, the time synchronization accuracy can be kept in the range of sub-microsecond with minimal network load and low demand on computing resources. Based on the OMNeT++ simulator, we build the PTP ordinary clock model and simulate the typical process of the PTP clock synchronization. The complete clock synchronization process and its performance are both analyzed and evaluated.**

Keywords- IEEE 1588; PTP; clock synchronization; OMNeT++; network simulation

I. INTRODUCTION

Timing requirements on networked measurement and control systems have become increasingly stringent, especially for those hard real time applications like substation automation [1], nuclear fusion control, mobile communication and advanced manufacturing plant, *et al*. Accurate time stamping of data at the source and precise triggering of the control actions are critical in those areas, since the measurement and control framework is composed of multiple distributed intelligent devices with various local time sources.

Currently, there are several technologies to achieve clock synchronization, such as Network Time Protocol (NTP), Simple Network Time Protocol (SNTP), IRIG-B and GPS. But few of them can both meet the high accuracy demand and low cost of deployment and maintenance. Among the existing time synchronization technologies, the IEEE 1588 PTP [2] has been recognized as the most prospective candidate for large, complicated industrial networks due to the striking features its provides: 1) It enables sub-microsecond clock synchronization, which makes it suitable for time critical applications; 2) It is implemented on UDP/IP, which makes it practical for networked measurement and control.

These features have drawn enormous attention over the past few years. Among them many efforts have been concentrated on protocol study, software design [3-4] and simulation. Because it is especially suitable for the performance analysis of large scale and heterogeneous systems as well as the behavior analysis of single modules like clock servo [5], timestamping mechanism and protocol stack [6].

The work of this paper mainly focuses on the modeling and simulation of the IEEE 1588 PTP ordinary clock, aiming to deeply understand the mechanism and process of the clock synchronization. The OMNeT++ [7], an open-source and object-oriented modular network simulator, is chosen as the simulation platform.

The remainder of this paper is organized as follows: A brief introduction to the IEEE 1588 PTP will be presented in section II, followed by the detailed descriptions of the PTP clock modeling in next section. In section IV, the behavioural simulation and analysis of the clock synchronization process will be discussed. The conclusions of this paper and the outlook for future work will be presented in the last section.

II. BRIEF DESCRIPTION OF THE IEEE 1588 PTP

A. *The PTP Clock Synchronization Mechanism*

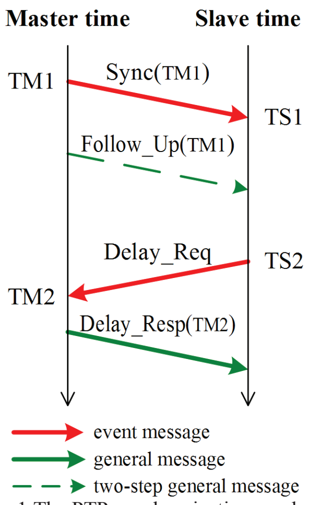

Figure 1 The PTP synchronization mechanism

The typical process of the PTP clock synchronization between a master clock and its slave nodes is based on a delay request-response mechanism, as is demonstrated in Fig. 1. It consists of the offset measurement and the delay measurement parts. The first part begins with the transmission of a multicast *Sync* message from the master node, while *TM1* is the message egress time referenced by the master clock. The slave node will immediately receive the *Sync* message at time *TS1* noted by the slave clock. The master node will then send a multicast *Follow_Up* message embedding *TM1* to all the slave nodes if one-step clock mode is not available. In the second part, the slave node sends a *Delay_Req* message to its master clock, while *TS2* is the egress time recorded at the slave clock. The master node receives this message and records its ingress time *TM2*. A *Delay_Resp* message containing *TM2* will be sent back by master node as soon as possible. Under the assumption that the transmission delay between the link of the master and the slave is symmetrical, then the clock offset between the master and the slave is derived as:

$$offset = \frac{(TS1 - TM1) - (TM2 - TS2)}{2} \tag{1}$$

B. Factors that Affect the Timestamp Accuracy

The digital time values referred in the above equation are obtained by means of taking timestamps. However, in real implementation, the accuracy of these timestamps is degraded by unavoidable timing fluctuations. Factors such as unpredictable latencies and jitter will inevitably be introduced due to message queuing, bit stream transfer and software operation. The time error may be up to several milliseconds at the worst case if the timestamps are generated at software layer rather than at the hardware level.

In short, the timestamps must be generated as close as possible to the physical layer, as the IEEE 1588 PTP standard specifies. By introducing the hardware assistance, it is feasible to generate timestamps between the data link layer (MAC) and the physical layer (PHY). The silicon chip can filter the PTP messages from packages in no time when they are passing through the medium independent interface (MII), recording their timestamps and place timestamps on the sending ones. Therefore, it is possible that the clock synchronization accuracy can be kept in the range of the sub-microsecond.

III. Description of the PTP Ordinary Clock Model

Fig. 2 releases the simulation model of the PTP ordinary clock, which follows a layer structure on both the hardware and software side. It is a compound model composed of several sub-modules connecting in some logical form. The clock model includes generic OSI layers, such as network layer (IP), transport layer (UDP), data link layer (MAC) and physical layer (PHY), all of them provide the fundamental support for the PTP clock synchronization. The model of these layers can be obtained from the INET Framework [8], an open-source network simulation package compatible with the OMNET++. The rest of the models need elaborated design and analysis, which compose the main work of this paper.

Among these PTP-related modules, module *PTP_Stack* and its application module are the core modules. Their main functions are providing basic control procedure for messages handling and event triggering, according to its clock state.

Module *PTP_DataSet* and *PTP_Port_DataSet* register all PTP defined data sets for normal running. Besides, the module *PTP_Port_DataSet* is also designed for the purpose of running

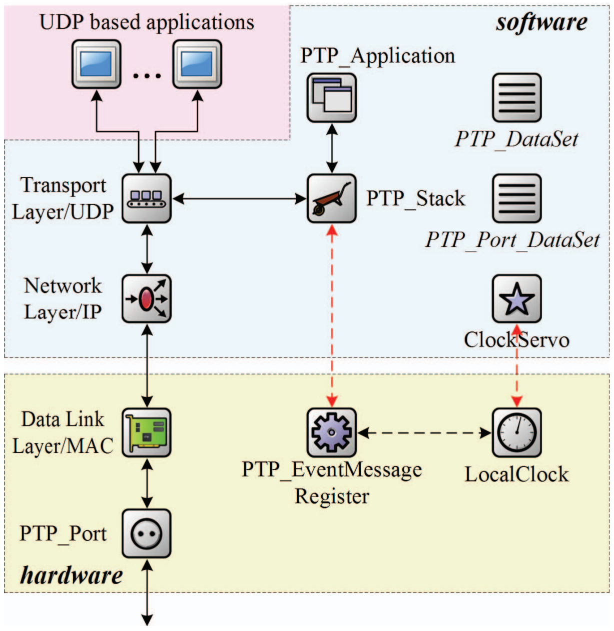

Figure 2 The simulation model of the PTP ordinary clock

the Best Master Clock (BMC) algorithm. The BMC algorithm is comprised of two parts, one is a data set comparison algorithm that determines which of two clock ports is better, the other is computing a recommendation state for each port involved. This avoids configurations with two masters or none in synchronization network. Module *PTP_Port* is built for detecting PTP messages. In each synchronization module *ClockServo* adjusts the local clock by using data extracted from the PTP messages and timestamps. These timestamps are all listed in module *PTP_EventMessageRegister*.

A. The Local Clock Model

The local clock refers to the local time source of the networked distributed device. It can be an atomic clock, a GPS or a crystal oscillator. The attention of this paper is focused on the modeling and adjustment of crystal oscillator, since it is widely adopted due to its low cost and capability of meeting most time accuracy demands. The PTP class *clockQuality* is one of the most essential data structure related to the local clock source. The items of this class are used to describe its attributes, stability and priority. These parameters are used in the BMC algorithm operation.

Papers [9-10] have deeply investigated the clock model and its behavior. The stability of the local clock is described by frequency skew (offset), with the dimensionless unit of *Part Per Million* (PPM). The frequency skew is defined as:

$$frequencySkew(t) = f_0 + Dt \tag{2}$$

Where the constant f_0 represents the static frequency offset at initial time, D is the frequency drift coefficient. In this paper, we assume that D is fixed. However, D is not always constant indeed. It is vulnerable to the environmental change such as temperature, moisture and other factors [11]. Then the local clock time error $e(t)$ at the reference time t is:

$$e(t) = e_0 + \int_0^t frequencySkew(t)dt + \varepsilon(t) \quad (3)$$

Where e_0 is the time offset at the reference time $t = 0$. The item $\varepsilon(t)$ represents the error generated during the implementation of clock synchronization. Factors such as systematic noise, clock jitter, clock granularity and other forms of random errors should be taken into account. The recursive form can be written as:

$$e(t) = e(t') + f_0(t - t') + 1/2D(t - t')^2 + \varepsilon(t) - \varepsilon(t') \quad (4)$$

Then the local time at the reference time t equals to:

$$localTime(t) = t + e(t) \quad (5)$$

B. The PTP ClcokServo

Briefly, the module *ClockServo* is responsible for two aspects of the clock adjustment. One is to calculate the time offset and correct it. The other is to estimate and compensate the clock frequency skew. Within each cycle, the time offset correction combined with the clock frequency skew adjustment is implemented. Paper [12] proposes a new approach of the clock frequency skew adjustment by introducing hardware measurement. In this paper, the clock frequency skew at cycle n is obtained from the following equation:

$$x[n] = \frac{syncIngressST_n - syncIngressST_{n-1}}{syncIngressMT_n - syncIngressMT_{n-1}} - 1 \quad (6)$$

Where $syncIngressST_n$ is the n-th Sync message ingress time measured at the slave node, while $syncIngressMT_n$ is the corresponding master clock time. For the correction of time offset and clock frequency skew, we respectively adopt a PI loop control to achieve the ideal performance.

IV. Simulation of the PTP Clock Synchronization

A. Simulation Configuration

As we mentioned in the first section, the focus of this paper is the behavior of the ordinary clock. Therefore, as Fig. 3 shows, the typical end-to-end (E2E) synchronization mechanism with only two ordinary clocks is considered in this simulation. Critical parameters related to the clock synchronization are presented in Fig. 4. The ordinary clocks communicate via UDP service with destination port 319 and 320 for event and general messages respectively. For the E2E delay mechanism, the multicast IPv4 address is 224.0.1.129.

B. Simulation Results and Process Analysis

Demonstrated in this case are the synchronization performances for local clocks with the accuracy of 1 ppm and

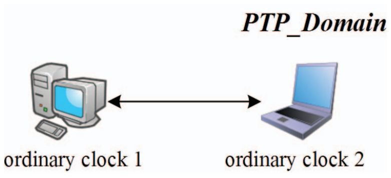

Figure 3 E2E simulation case

parameters	***configuration***	
	ordinary clock 1	***ordinary clock 2***
two step clock	true	
slave only	false	true
Announce message interval	1 s	
Sync message interval	1 s	
delay mechanism	E2E	
time source	reference time	oscillator (D=1e-10)
data rate	100*Mbps*	

Figure 4 Simulation environment configurations

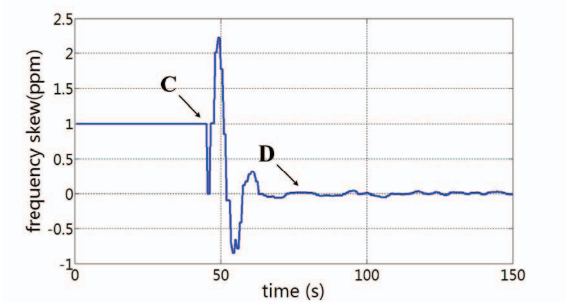

Figure 5 Clock frequency skew correction (1 ppm)

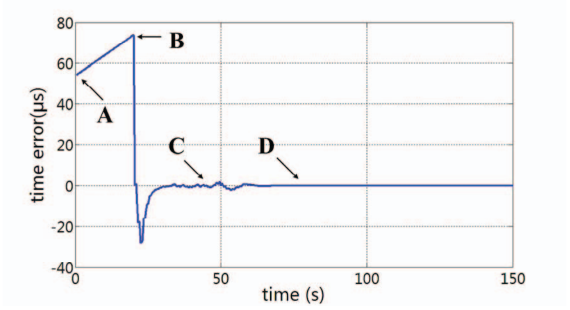

Figure 6 Performance of clock synchronization (1 ppm)

100 ppm. The synchronization performances of these clocks are shown in Fig. 6 and Fig. 9, while Fig. 7 and Fig. 10 are referred to the corresponding timer error histograms. The synchronization process can be divided into four sections, as shown in Fig. 5 and Fig. 6, the dynamic performance plot.

1) Initialization

Position ‘A’ marks the moment of system power up. Normally, an ordinary clock will first initialize its data sets and then enter ‘listening’ state. All active nodes in the PTP domain will operate BMC algorithm to determine their best clock states. If an ordinary clock recognizes itself as slave, it will be ready to handle the synchronization messages from its master. Time error boosts drastically during this time because the synchronization has not really worked yet. This section lasts approximately 20 seconds.

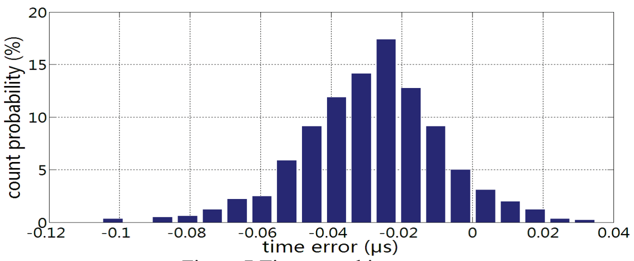

Figure 7 Time error histogram

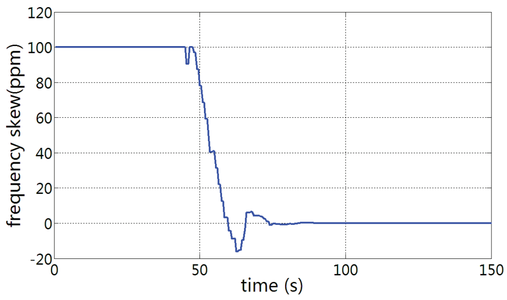

Figure 8 Clock frequency skew correction (100 ppm)

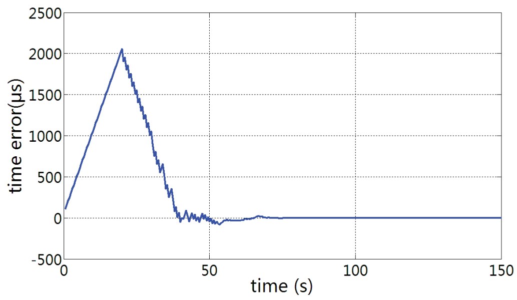

Figure 9 Performance of clock synchronization (100 ppm)

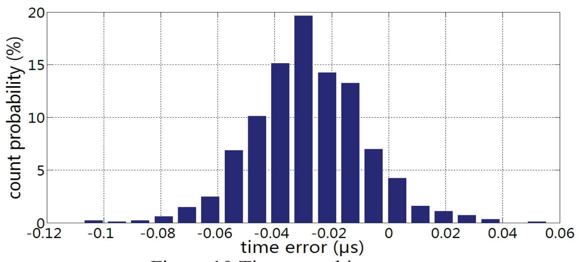

Figure 10 Time error histogram

2) Compensation for main time offset and estimation of clock frequency skew.

The synchronization mechanism starts to function from point 'B'. Much of the time error is successfully compensated by applying PI control at each synchronization cycle. However, it can not be totally eliminated since the frequency skew is the main source of the time error.

3) Clock frequency skew correction. After estimating the clock frequency skew, point 'C' is the beginning of the frequency skew correction process (see Fig. 5).

4) Normal state of synchronisation. As the frequency skew is well controlled at position 'D', the clock synchronization process is finally in its real stable status.

Two conclusions can be drawn from the simulation results:

1) Due to the high accurate hardware timestamps, much of the time error can be removed quickly. 2) By the use of the clock frequency skew correction, time error can be further suppressed to the range of less than a hundred nanoseconds, regardless of the quality of the clock (1 or 100 ppm).

V. Conclusion and Future Work

In this paper, we present an OMNeT++ based PTP ordinary clock and discuss the details of each module. The simulation provides us an efficient approach to analyze and evaluate the process of the clock synchronization. It also shows the high synchronization performance and fast convergence. But it is far from collecting enough data to fully investigate the PTP performance due to lack of more complex experiments. Cross-traffic or network with large number of ordinary clocks connected by switches may incur asymmetrical delay, degrading synchronization performance. This is also viewed as part of PTP synchronization performance evaluation in the future. It is necessary to model other basic PTP devices such as end-to-end and peer-to-peer transparent clock to revaluate the performance with different synchronization mechanism.

References

[1] F. Steinhauser, C. Riesch, M. Rudigier, IEEE 1588 for Time Synchronization of Devices in the Electric Power Industry, in: 2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS 2010), 27 Sept.-1 Oct. 2010. (IEEE, Piscataway, NJ, USA, 2010), 6 pp.

[2] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Std 1588 (2008) xvii+269.

[3] L. Kang, E. Song, Object-oriented model for IEEE 1588 standard, in: 2007 International IEEE Symposium on Precision Clock Synchronization (ISPCS) for Measurement, Control and Communication, 1-3 Oct. 2007. (IEEE, Piscataway, NJ, USA, 2007), 7-12.

[4] E.Y. Song, K. Lee, An application framework for the IEEE 1588 standard, in: 2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 22-26 Sept. 2008. (IEEE, Piscataway, NJ, USA, 2008), 23-28.

[5] G. Giorgi, C. Narduzzi, Modeling and simulation analysis of PTP clock servo, in: 2007 International IEEE Symposium on Precision Clock Synchronization (ISPCS) for Measurement, Control and Communication, 1-3 Oct. 2007. (IEEE, Piscataway, NJ, USA, 2007), 155-161.

[6] K. Correll, N. Barendt, M. Branicky, Design considerations for software only implementations of the IEEE 1588 precision time protocol, in. (Citeseer, 2005).

[7] OMNeT++ release 4.0 http://www.omnetpp.org/

[8] INET Framework for OMNeT++/OMNEST release 2010-03-23. http://inet.omnetpp.org/.

[9] D.W. Allan, Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators, IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, UFFC-34 (1987) 647-654.

[10] G. Giorgi, C. Narduzzi, Performance analysis of Kalman filter-based clock synchronization in IEEE 1588 networks, in: 2009 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication. ISPCS 2009, 12-16 Oct. 2009. (IEEE, Piscataway, NJ, USA, 2009), 6 pp.

[11] Precision oscillators: dependence of frequency on temperature, humidity and pressure, in: Proceedings of the 1992 IEEE Frequency Control Symposium (Cat. No.92CH3083-3), 27-29 May 1992. (IEEE, New York, NY, USA, 1992), 782-793.

[12] T. Neagoe, M. Hamdi, V. Cristea, Frequency compensated, hardware IEEE-1588 implementation, in: 2006 IEEE International Symposium on Industrial Electronics, 9-13 July 2006. (IEEE, Piscataway, NJ, USA, 2006), 240-245.