Packet Delay Estimation Algorithm against FIFO Waiting for Asymmetric IEEE 1588 System

Lei Chen
Beijing Institute of Tracking & Telecommunication Technology, Beijing, China
Email: chenlei0831@126.com

Tianlin Zhu
Beijing Institute of Tracking & Telecommunication Technology, Beijing, China
Email: zhutianlin@yahoo.com.cn

***Abstract*—Performance in IEEE 1588 synchronization depends on several related factors. Among them, the symmetry of packet delay is the most basic one. But most existing networks could not provide symmetry packet delay between master and slave clocks. From research we found that, FIFO waiting during packet transmitting is one of the main reasons that lead the asymmetry. This paper puts forward a packet delay estimation algorithm to select those "lucky packets" which survived from FIFO waiting, attenuating the FIFO waiting effects on IEEE 1588 synchronization. Verified by some meaningful tests, compared with no optimization, in an asymmetry network, the accuracy of packet delay estimation increases almost 25ns with stability increasing almost 2 order, and the accuracy of IEEE 1588 synchronization increases more than one-times with stability increasing almost four-times.**

***Index Terms*—IEEE 1588, symmetry of packet delay, FIFO waiting, packet delay variation (PDV), lucky delay**

I. INTRODUCTION

With the distributed trend development of space measurement and control systems, and the application of integrative information system, the synchronization among the terminals in a system becomes more and more important. However, the traditional time synchronization based on B-code[1] transmitted in specified line could not fit the distributed character, and now the already used network time synchronization based on NTP (Network Time Protocol) could only achieve poor accuracy in millisecond, which could not meet the accuracy requirements. To provide effective solutions and realize high accuracy time synchronization based on network, IEEE published IEEE 1588 standard first in 2002 and the modified version in 2008. IEEE 1588 standard [2] defines a kind of precision clock synchronization protocol for networked measurement and control systems, which is called PTP (Precision Time Protocol).

In a network enhanced with IEEE 1588 function, such as IEEE 1588 packet transmitted first, modification and compensation in packet delay and so on, the IEEE 1588 packet delay usually would not vary, which protect the symmetry of packet delay between master-to-slave and slave-to-master. Thus the synchronization could achieve sub-nanosecond accuracy even nanosecond level with hardware support [3].

However, most of existing network haven't support IEEE 1588 yet and the IEEE 1588 packet transmission has to follow the FIFO (First In First Out) principle, which would make some packets transmitted delay, that is called FIFO waiting. The uncertainty of FIFO waiting would lead packet delay variation (PDV)[4] and make master-to-slave and slave-to-master packet delays asymmetric and damage the performance of IEEE 1588 synchronization.

To attenuate the influence of FIFO waiting, multi methods have been tried, such as a kind of clock servo architecture based on filtering proposed in [5], some adaptive filtering design introduced in [6] and [7] and so on. But these solutions are all too complex with uncontrolled stability. Reference [2] proposed using boundary clock and transparent clock which can attenuate the damage radically. But it needs mass alteration with staggering cost. So there is need for a low-cost and stable solution which can realize high-accuracy and high-stability IEEE 1588 synchronization in asymmetric network.

This paper focuses to study the FIFO waiting and its influence on IEEE 1588 synchronization. Based on the analysis, we proposed a kind of packet delay estimation algorithm against FIFO waiting. Through analysis of the relationship between the parameters and the performance indexes of IEEE 1588 synchronization, we derived its boundary conditions and verified its effectivity with some meaningful tests.

II. IEEE 1588 SYNCHRONIZATION PRINCIPLE

IEEE 1588 has provided effective solutions to realize high-accuracy synchronization based on network. Following the "delay require-response" principle, the protocol transfers packets periodically to provide precision timestamps for estimating the offset from master-to-slave in IEEE 1588 slave clocks. According to the estimated offset, slave clocks would adjust its local

Manuscript received November 20, 2013; November 30, 2013; accepted January 31, 2014.
National High-tech R&D (863) Program of China (2011AA0474).

doi:10.4304/jcp.9.9.2075-2081

clock through servo system. After several periodical adjustment, slave clocks could achieve synchronization with master clock in the end. The basic diagram of IEEE 1588 synchronization is shown as Fig.1.

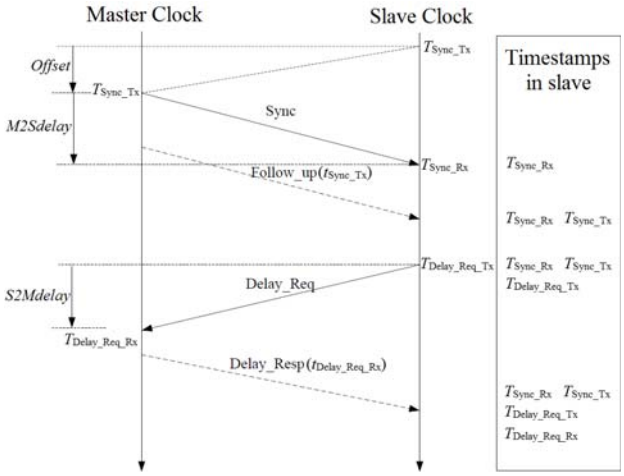


Figure 1. Diagram of IEEE 1588 synchronization.

The process of IEEE 1588 synchronization could be divided into two steps: clock offset measurement and packet transmission delay measurement.

In the process of clock offset measurement, master clock would send synchronization packet "Sync" to slave clock and meanwhile mark the precision leaving time of packet "Sync" as T_{Sync_Tx}. And then, slave clock would mark the precision arriving time of packet "Sync" as T_{Sync_Rx}. Since the leaving time information T_{Sync_Tx} could not be included into the packet "Sync" precisely, IEEE 1588 defines packet "Follow_up" to transmit the leaving time information of packet "Sync" precisely. If we define the packet transmission delay from master to slave as M2Sdelay, and the offset between master and slave as Offset, we can calculate the offset from (1).

$$Offset = T_{Sync\_Rx} - T_{Sync\_Tx} - M2Sdelay \quad (1)$$

The unknown M2Sdelay in (1) could be calculated by packet transmission delay measurement. In the process of packet transmission delay measurement, slave clock would firstly send delay requirement packet "Delay_req" to master clock and meanwhile mark its precision leaving time as $T_{Delay\_Req\_Tx}$. And then, master clock would mark the precision arriving time of packet "Delay_req" as $T_{Delay\_Req\_Rx}$. In the end, master clock would send the arriving time information back to slave clock with packet "Delay_resp". If we define the packet transmission delay from slave to master as S2Mdelay, then we can get

$$S2Mdelay = T_{Delay\_Req\_Rx} - T_{Delay\_Req\_Tx} + Offset \quad (2)$$

Here, IEEE 1588 has a basic assumption that the packet transmission delay from master to slave is the same as from slave to master, that is to say

$$M2Sdelay = S2Mdelay \quad (3)$$

Based on this assumption, we can get the offset

$$Offset = \frac{(T_{Sync\_Rx} - T_{Sync\_Tx}) - (T_{Delay\_Req\_Rx} - T_{Delay\_Req\_Tx})}{2} \quad (4)$$

So from (4) we can get that the accuracy of estimated offset is nearly related to the symmetry of packet delay. If the assumption that packet delay is symmetry could be meet, the estimated offset would reflect the real offset. But if the assumption could not be meet, the synchronization based on estimated offset would be damaged.

II. FIFO WAITING

To satisfy the assumption of symmetry, IEEE 1588 defined boundary clock and transparent clock to realize IEEE 1588 packet transmitted first, modification and compensation in packet delay. However, most of existing network haven't support IEEE 1588 yet, so PTP packet transmission had to follow the FIFO principle, which would make some packets transmitted delay when other packets cutting in, and this is the FIFO waiting.

FIFO waiting is just as shown in Fig.2. Firstly, master clock would sent PTP packets 1,2,3,4,5…periodically to slave with interval Δt. During the interval of packets transmitted across switches without IEEE 1588 function, if there was other packets cutting in, some unfortunate PTP packets would fall behind in the transmitting queue and be transmitted with uncertainty delay. Just like the packet 2 in Fig.2, the interval is Δt when it leaves from master, but effected by FIFO waiting when across switches, the interval becomes Δt_1. Since the uncertainty of FIFO waiting, the delay caused by FIFO waiting is hard to be compensated.

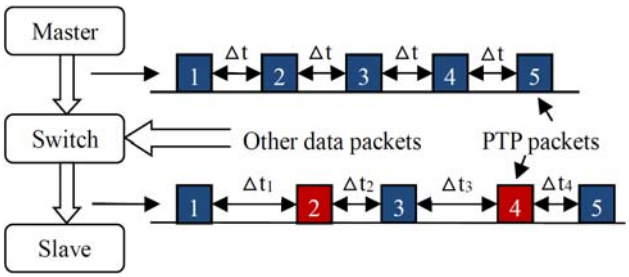


Figure 2. FIFO waiting in transmission of PTP packets

The packet delay with uncertainty variation could not be symmetry again between master-to-slave delay and slave-to-master delay. So the offset calculated by (4) could not reflect the real offset. All these would finally damage IEEE 1588 synchronization performance, as Fig.3 and 4 shows.

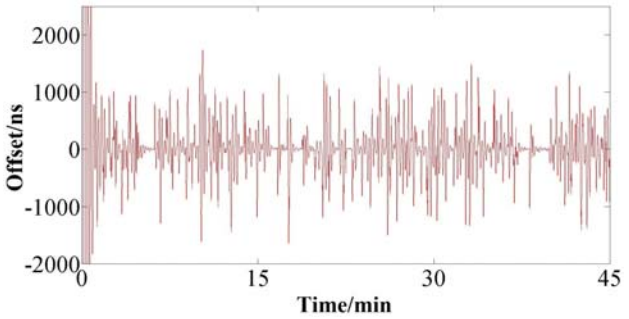


Figure 3. Damaged IEEE 1588 synchronization performance.

From Fig.3, we can see that, since there was PDV caused by FIFO waiting, the (3) assumption could not be

meet and the offset estimation based on this assumption would be accurate. And the so called adjustment signal according to this inaccurate estimation outputted by followed servo system could not adjust the local clock, but contrarily, would damage the synchronization. Fig.4 shows the statistical analysis of the synchronization in Fig.3. If we regard that the offset between negative 100 nanoseconds and positive 100 nanoseconds is effective synchronization, then this proportion is only 40.85% from Fig.4 and could not meet the accuracy requirement.

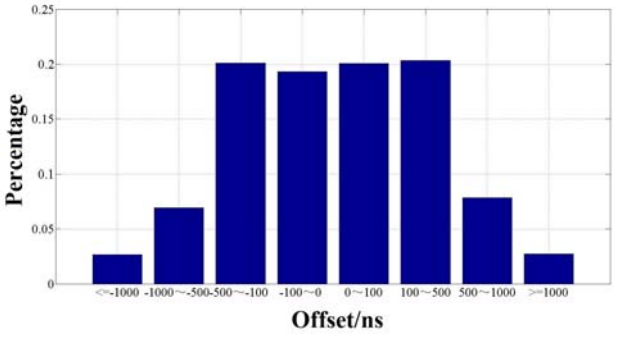


Figure 4. Statistical analysis of damaged IEEE 1588 synchronization.

III. Packet Delay Estimation

From research we found that, packet delay would varied from minimum physical delay to maximum delay when they across a network equipment in networks without IEEE 1588 function and meanwhile the minimum delay is not a rare event[8]. As the packet 3 in Fig.2, it survived from being cut in and avoiding transmitted delay. The interval between packet 2 and packet 3 is almost Δt. So the sum of packet delay from master to slave would be minimum in probability. Based on this, we proposed a packet delay estimation with minimum FIFO waiting algorithm against PDV. This algorithm try to select those lucky packets surviving from FIFO waiting in an acceptable time interval. These selected lucky packets could meet the assumption (3) since that there is nearly no PDV in them.

The PTP packets periodic alternation is shown as Fig.5. $\Delta t_{Sync[i]}$ and $\Delta t_{Delay\_Req[i]}$ are the interval of Sync[i] and Delay_Req[i] separately from their former packets. $t_{Sync[i]}$ and $\Delta t_{Delay\_Req[i]}$ should be the same with synchronization period Δt_{Sync} and delay requirement period Δt_{Delay_Req} separately in theoretical without FIFO waiting. But if there was FIFO waiting, they would be not the same.

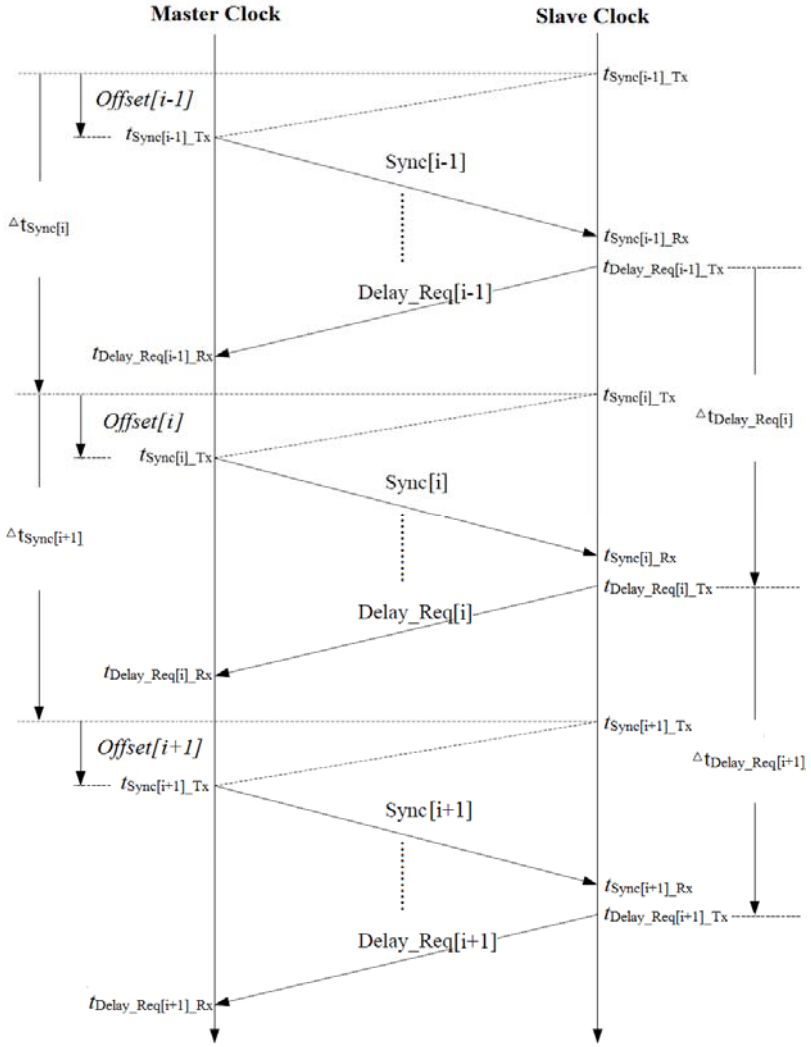


Figure 5. Alternation of PTP packets between master and slave.

Packet delay estimation with minimum FIFO waiting receives new information Sync[i] and Delay_Req[i] in stack and then calculates time interval $\Delta t_{Sync[i]}$ and $\Delta t_{Delay\_Req[i]}$. The stack would keep receiving new information and its output would not be changed until it is full. When the stack is full, the lucky packets would be selected according to the principle. With the timestamps in lucky packets, the mean path delay would be calculated and outputted, meanwhile the stack would be cleared to wait for new coming information. If following the selecting principle, no lucky packets were selected, then the output would be the same as last output. The diagram of this algorithm is shown as Fig.6.

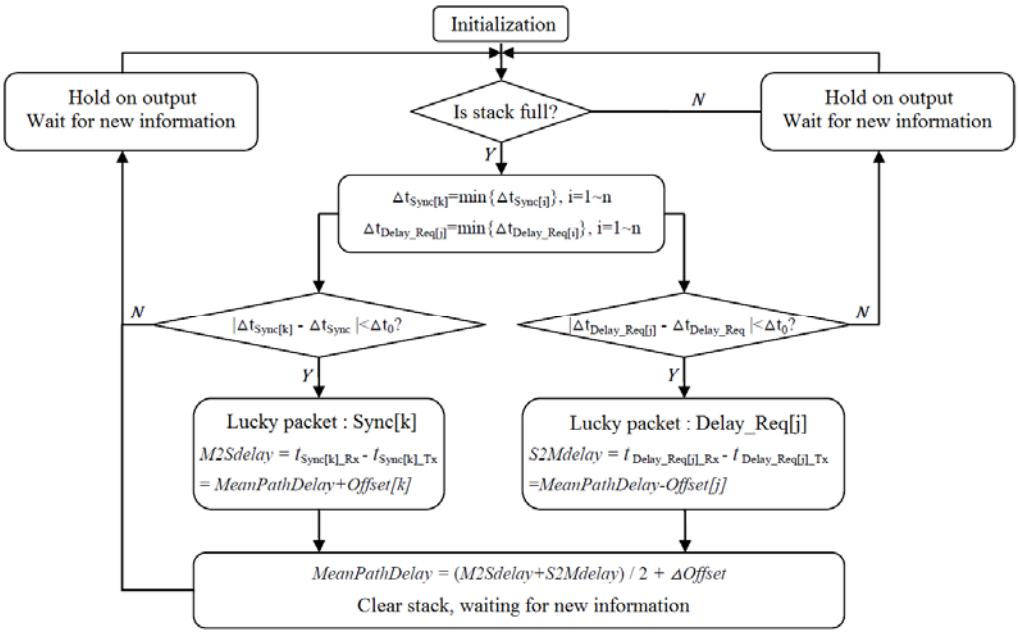


Figure 6. Diagram of packet delay estimation against FIFO waiting

With the interference of other random noise, the two variables $\Delta t_{Sync[i]}$ and $\Delta t_{Delay\_Req[i]}$ would not be completely the same with the constant Δt_{Sync} and Δt_{Delay_Req} even without FIFO waiting. So the selecting principle is comparing the variable with the corresponding constant as follows:

(1) If the absolute D-value of the variable and its corresponding constant is greater than a fixed threshold Δt_0, then we consider the packet with FIFO waiting.

(2) If the absolute D-value of the variable and its corresponding constant is smaller than a fixed threshold Δt_0, then we consider the packet as lucky packet.

But since that the selected "synchronization lucky packets Sync[k]" and "delay-requirement lucky packets Delay_Req [j]" may not from the same period, there would be little offset, that is *ΔOffset* in Fig.6, in mean path delay calculated based on the two lucky packets. *ΔOffset* is decided by the interval number of synchronization period:

(1) If the two lucky packets were from the same period, k=j, then *ΔOffset*=0;

(2) If the interval number was too big, *ΔOffset* would not be acceptable.

The interval number is related to stack depth "n". A big "n" would make good selecting but may cause an unacceptable *ΔOffset*. While, a small "n" would make selecting meaningless and damage system stability.

Meanwhile, the threshold Δt_0 is also a very important parameter for this algorithm. It should be decided by synchronization requirements and system configuration. For slave clocks with high-stability oscillators, Δt_0 could be set small properly to make good selecting. Although a small threshold needs a large interval for selecting, IEEE 1588 synchronization accuracy would not be damaged, for the good performance in keeping time in slave. But for slave clocks with low-stability oscillators, Δt_0 should be set large properly to decrease selecting time and time error caused by long selecting time.

Since that this algorithm needs some necessary time to select lucky packets. However, long selecting time may cause poor synchronization, so the boundary condition of this algorithm is decided by mean selecting time interval.

Assuming that the transmission of a PTP packet from master to slave passes through N_{sw} switches, and the line utilization of cross traffic may lead FIFO waiting at the *i*-th switch is $\alpha_{i,\,(i=1\sim Nsw)}$, then the probability that the packet survived from FIFO waiting is:

$$P_{no\_PDV} = \prod_{i=1}^{N_{sw}} (1-\alpha_i) \tag{5}$$

Considering the depth of stack "n" and synchronization interval T_{sync}, the mean selecting interval where a lucky packet could be selected is obtained from

$$T_{mean} = \frac{T_{sync}}{1-(1-P_{no\_PDV})^{n-1}} \tag{6}$$

Based on assumption that the line utilization of cross traffic at each switch is the same and temporally constant α_0, the mean interval T_{mean} calculated from (6) is shown in Fig.7.

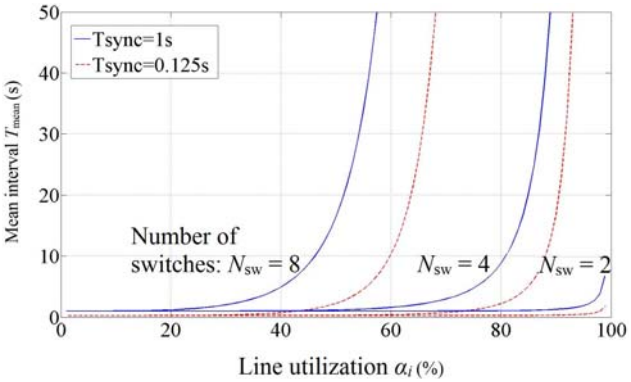


Figure 7. The mean searching interval needed by packet delay estimation against FIFO waiting.

From Fig.7 we can see that the mean interval would increase with the line utilization increasing and steeply when line utilization reaching some point. Also the number of switches N_{sw} and synchronization interval T_{sync} play an important role in deciding mean interval. In the same environment, the mean interval would increase as the number of switches increases.

The tolerance of mean selecting interval of a PTP synchronization system mainly depends on the performance of oscillator. So we should carefully configure the number of switches N_{sw}, synchronization period T_{sync} and the selecting threshold Δt_0 to protect the algorithm is effective when running PTP synchronization.

IV. Test & Verification

To verify the effectiveness of this algorithm, we built on a real test environment as shown in Fig.8 based on "ITU-T G.8261 Test Case Profiles for 1588V2 Testing (TestCase12)" in [9]. The IEEE 1588 master and slave clocks are both developed based on STM32F107[10] in hardware and open source code PTPd in software. Some pertinent modifications are mainly in the file "servo.c" and the PTP clock data structure to realize this algorithm.

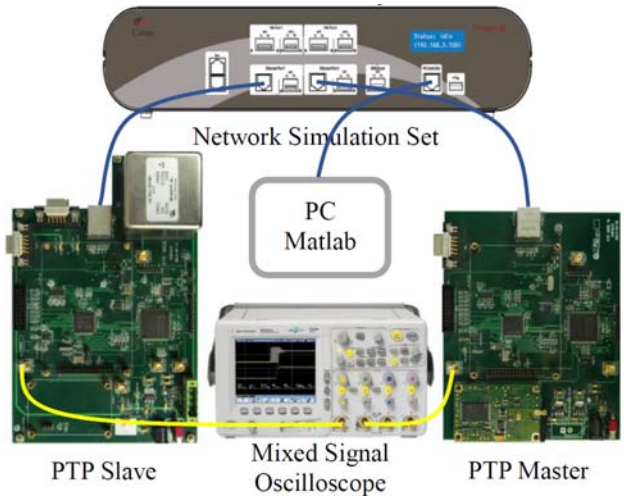


Figure 8. Schematic diagram of simulation and verification.

In Fig.8 the network simulation set is "paragon-X" from Calnex Inc., used to test PTP synchronization performance. The detail test follows "ITU-T G.8261[11] Test Case Profiles for 1588V2 Testing (TestCase12)" in [9]. The data statistical analysis is done by MATLAB on PC with the data sent back from slave clock. The parameters are as listed in Tab.1.

TABLE 1.
THE MEAN SEARCHING INTERVAL NEEDED BY PACKET

T_{sync}	T_{Delay_Req}	α_{M2S}	α_{S2M}	n	Δt_0
1s	1s	20%	80%	20	50ns

Fig.9 shows the output of packet delay estimation without optimization compared with optimization. From it we can see that the packet delay estimation is disturbed by PDV seriously, and the mean estimation without optimization $\overline{MeanPathDelay}$ =20523.22ns, the standard deviation $\sigma_{MeanPathDelay}$ =961.62ns. Fig.10 magnifies the output of packet delay estimation with optimization. From it we can see that PDV caused by FIFO waiting could be filtered out, and the mean estimation with optimization $\overline{MeanPathDelay}$ =20550.15ns, the standard deviation $\sigma_{MeanPathDelay}$ =4.26ns, the needed mean selecting interval is about two synchronization periods.

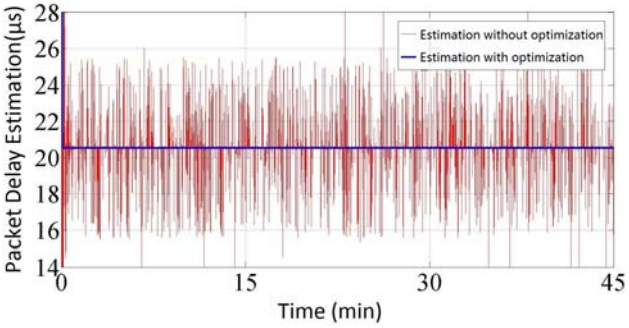


Figure 9. Output comparison of packet delay estimation.

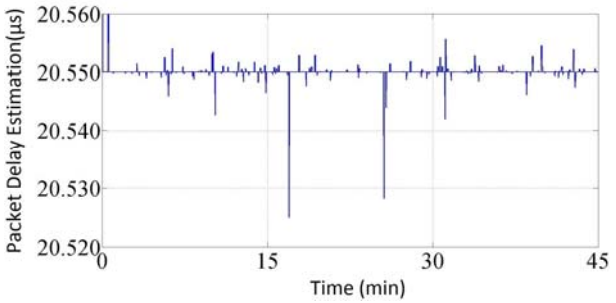


Figure 10. Magnified display of packet delay estimation output with optimization.

From statistical real test, we can get the minimum physical delay is about 20550ns. So compared with the test results, we can see that packet delay estimation with min-FIFO waiting is more accuracy and stable. The mean accuracy is improved with 25ns, and stability is advanced by 2 order of magnitude. Thus, the packet delay estimation based on lucky packets could be meet the basic assumption (3), in an asymmetric network.

And the offset calculated by (4) could be reflect the real offset, letting the servo system could output more accuracy and stable adjustment signals, and finally

improving the synchronization performance. The synchronization performance after optimization is shown as Fig.11. Compared with Fig.3, the improvements in both accuracy and stability are obvious. Fig.12 shows the statistical analysis of the synchronization in Fig.11. More detail data comparison is shown in Tab.2.

TABLE 2.
THE MEAN SEARCHING INTERVAL NEEDED BY PACKET

Percentage in ±100ns Without optimization	Percentage in ±100ns With optimization	Standard deviation Without optimization	Standard deviation With optimization
40.85%	87.83%	376.44ns	90.16ns

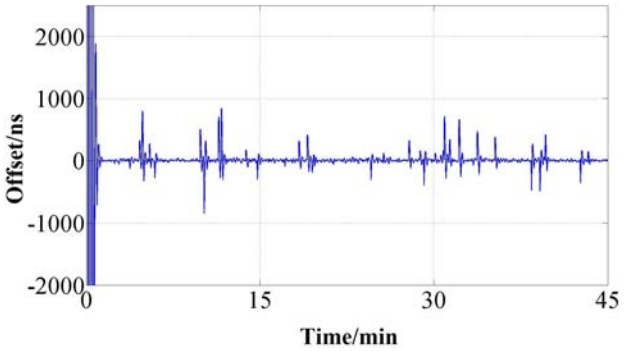


Figure 11. Optimized IEEE 1588 synchronization performance.

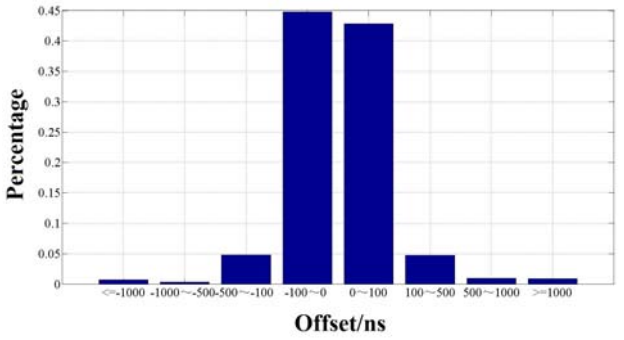


Figure 12. Statistical analysis of optimized IEEE 1588 synchronization performance.

V. CONCLUTION

To solve the problem that the packet delay is not symmetric between master and slave in an IEEE 1588 network, we proposed a packet delay estimation against FIFO waiting algorithm based on stack-style selecting. Analyzing the relationship between performance index, such as oscillator's stability and synchronization accuracy requirement and so on, and related parameters in this algorithm, we also derived the boundary condition of this algorithm. Based on the real test environment built on paragon-X with network traffic model defined by ITU-T G.8261, and PTP master and slave clocks, we set different line utilizations in forward direction and reverse direction to simulate the asymmetry caused by PDV. Finally verified by real test, the mean accuracy is improved with 25ns, and stability is advanced by 2 order of magnitude. So the algorithm is proved effective.

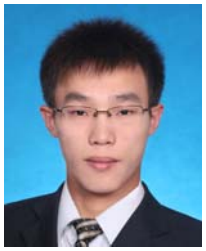
The packet delay estimation against FIFO waiting algorithm can make packet delay estimation satisfy the assumption that the packet delay is symmetric, even in an asymmetric network. It attenuated the influence of PDV caused by FIFO waiting on IEEE 1588 synchronization performance, improving the accuracy and stability. The most important is that this algorithm provides a technology reference for IEEE 1588 application in existing networks. In future, we would make a further research on such as clock servo, the effect of oscillator's quality on IEEE 1588 and so on.

ACKNOWLEDGMENT

The authors wish to thank Senior Engineer Liu Feng, Wang Wei and Hou Tongqiang, Assistant Engineer Chen Xudong for their support on the real test, the timing system design lab and the communication lab in BITTT for providing test environment, the IEEE for providing this template.

REFERENCES

[1] Tong Baorun. Timing System[M]. Nation Defence Industry Press. 2003.09 [China, pp.206-214, 2003]
[2] IEEE Std 1588-2008. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems[S]. 2008
[3] Wei Feng, Sun Wenjie. Precise time stamping method for IEEE1588 clock synchronization message. Chinese Journal of Scientific Instrument. Vol. 30, No. 1, pp.162-169, 2009.01 [China, pp.162-169, 2009]
[4] Takahide Murakami, Yukio Horiuchi. Improvement of Synchronization Accuracy in IEEE 1588 Using a Queuing Estimation Method[C]. Proc. IEEE Int. IEEE Symp. Precision Clock Synchronization for Meas., Control and Commun.(ISPCS). 2009
[5] Kendall Correll, Nick Barendt, Michael Branicky. Design Considerations for Software Only Implementations of the IEEE 1588 Precision Time Protocol[C]. Proc. IEEE Int. IEEE Symp. Precision Clock Synchronization for Meas., Control and Commun.(ISPCS). 2005
[6] Chen Lei, Zhu Tianlin. An Adaptive Filtering Algorithm With Packet Delay Variation For IEEE1588 servo system[C]. Proc. Int Conf. on Electronic Measurement & Instruments (ICEMI). 2013.08
[7] Takahide Murakami, Yukio Horiuchi, Kosuke Nishimura. A Packet Filtering Mechanism with a Packet Delay Distribution Estimation Function for IEEE 1588 Time Synchronization in a Congested Network[C]. Proc. IEEE Int. IEEE Symp. Precision Clock Synchronization for Meas., Control and Commun.(ISPCS). 2011.
[8] Patrick O'Farrell, David Rosselot. IEEE1588 synchronization in standard network with DP83640[EB]. NI- AN1963. 2009.05
[9] Calnex Solution Ltd-AN. Testing IEEE1588 V2 slave clocks (CX5003)[EB]. 2012.10
[10] ST-AN3411. IEEE 1588 precision time protocol demonstration for STM32F107 connectivity line microcontroller[EB]. 2011.05
[11] ITU-T Recommendation G.8261 Amendment 1. Network Jitter Limits for the Synchronous Ethernet Equipment Clock Interface and Other Clarifications. 2010.07



Lei Chen was born in Tieling, Liaoning Province, China, in 1988. He received BS in electronic engineering from Shanghai Jiaotong University, China, in 2011.

He is a MS candidate in Beijing Institute of Tracking and Telecommunication Technology now. His research interests include time synchronization using IEEE1588 and design of integrated systems.

Mr. Chen has been awarded as the excellent academic report first prize in the 11th International Conference on Electronic Measurement & Instruments (ICEMI) in Harbin, on August 19th 2013.

Tianlin Zhu was born in Yicheng, China, in 1967. He received BS from Huazhong University of Science and Technology. Now, Mr. Zhu is the section chief of Science&Technology Section in Beijing Institute of Tracking & Telecommunication Technology (BITTT).