Research of the Precision Clock Synchronization Based on IEEE 1588

Weikang Qian
University of Shanghai for Science Technology/School of Optical-Electrical and Computer Engineering, Shanghai, China
E-mail: elelsec@usst.edu.cn

Chao Guo and Junhui Mei
University of Shanghai for Science Technology/School of Optical-Electrical and Computer Engineering, Shanghai, China
E-mail: guochaohappy123@163.com may.0714@163.com

***Abstract*—In distributed measurement and control system, it is the key to need a global time to determine the order of events happened in system, to achieve the effectiveness of monitoring and control data which is various information transmissions or operation state， therefore the clock synchronization technology is required to synchronize the every slave subsystem. IEEE 1588 precision clock synchronization protocol can effectively solve the time synchronization problem in distributed measurement and control system. First, the basic principles of the IEEE 1588 precision clock synchronization protocol and related program flow are introduced in this article. Second, two algorithms applied to clock synchronization including DS comparison algorithm and State decision algorithm was further proposed and discussed. Meanwhile, the clock synchronization with PTP (Precision Time Protocol) between two devices communication was implemented on the platform of embedded Linux operating system. Finally test showed the accuracy of synchronization was reached less than dozens of microseconds.**

***Index terms*—IEEE1588 ; PTP; clock synchronization ; Linux**

I. INTRODUCTION

The development of interface bus for electronic test instruments had been through the GPIB (General Purpose Interface Bus) bus in the 1970s, VXI (VME bus eXtensions for Instrumentation) bus in the 1980s and PXI (PCI eXtensions for Instrumentation) bus in the 1990s. In 2004，Agilent cooperated projects with VXI Technology company to establish initiatively LXI alliance, and put forward a new generation of modular instrument bus—LXI combined with GPIB and VXI advantages. LXI means that instruments conforming LXI can be divided into three basic classes “A”, “B” and “C” according to the difference of synchronous and trigger mode, in which:

“C” is the basic grade of LXI devices, it’s required to provide LAN and Web interface of LXI- standards compliant and with the properties of simple structure, lower price and smaller size, it can be equivalent to the foundation capabilities of current CPIB system such as programming control ability by LAN, and instruments compatible characteristics.

“B” can involve all the function of the class “C” and support the IEEE 1588 precise time synchronization protocol standard. IEEE 1588 interface allows the device to perform the GPIB equivalent trigger and same or higher synchronization accuracy.

In addition to all the capabilities of the class “B”, “A” is advanced type with the hardware trigger mechanism to start trigger bus. As the development of network technology, the time synchronization precision of the measurement technology and control systems in a distributed network become more and more prominent problem to be solved. Because the requirement of system speed, the precision of the synchronization has also become an important technical index, and how to provide reliable clock service in the network is an important research topic.

The IEEE 1588 standard is known as "Precision Clock Synchronization Protocol for Networked Measurement and Control Systems" or "PTP" for short, and got the approval of the IEEE in 2002, released V2.2 in 2008. PTP defines a procedure allowing many spatially distributed real-time clocks to be synchronized through network (normally Ethernet). The technology behind the IEEE 1588 standard was originally developed by Agilent and was used for distributed measuring and control tasks. The challenge was to synchronize networked measuring devices with each other in terms of time so that they are able to record measured values and provide them with a precise system time stamp. Based on this time stamp, the measured values can then be correlated with each other. The protocol achieved the equipment clock synchronization in network through the combination of software and hardware , its precision can be reached to ns level, not only can realize the time synchronization, but also can offer frequency synchronization, and can basically meet the requirements of network measurement and control system. In view of the embedded operating system is stable, open source, portability, etc., embedded systems should include multiple processors to enhance processing ability, and also need application-specific processor to guarantee real-time requirement[1].

doi:10.4304/jsw.8.2.451-458

The Linux kernel was chosen because it is an attractive alternative for a large spectrum of applications, from laptops and desktops to big servers. The widespread use of Linux is consequence of all the advantages offered by this modern general purpose operating system, such as a multitask environment, communication protocol stacks, graphical resources, wide hardware support, code stability, continuous evolution and constant modernization for elimination of bugs. Another advantage of the use of Linux is the possibility of studying it, to alter and to do any kind of adjustment it may be necessary in order to adapt it to a certain embedded application [2], this paper described to achieve a precision clock synchronization protocol with using the Linux embedded operating system platform.

II. ANALYSIS OF IEEE 1588 PRECISION CLOCK SYNCHRONIZATION

Clock model is as shown in Fig.1 including the local clock system, the protocol state machine conversion, clock settings and the events interface, general interface, events message and other modules.

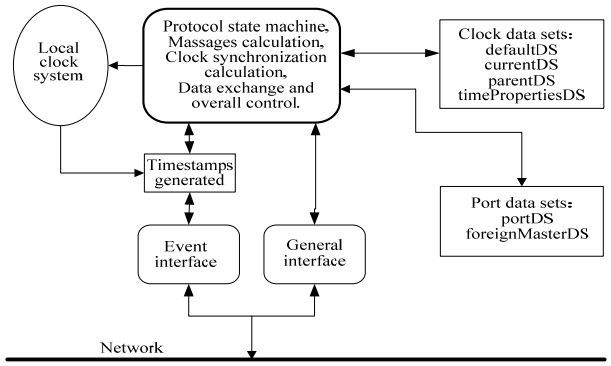

Figure 1. clock mode

Events interface module: used to send and receive event messages, generate time stamp based on the time-stamp generator of the local clock.

General interface module: used to send and receive ordinary messages.

Event messages include sync, Delay_Req, Pdelay_req, Pdelay_resp. Announce, Follow_up, Delay_resp, Pdelay_resp_follow_up are General messages.

The Sync, Delay_Reg, Follow_Up, and Delay_Resp messages are used to generate and communicate the timing information needed to synchronize ordinary and boundary clocks using the delay request-response mechanism. The Pdelay_Req, Pdelay_Resp, and Pdelay-Resp-Follow-Up messages are used to measure the link delay between two clock ports implementing the peer delay mechanism. The link delay is used to correct timing information in Sync and Follow_up messages in systems composed of peer-to-peer transparent clocks. Ordinary and boundary clocks that implement the peer delay mechanism can synchronize using the measured link delays and the information in the Sync and Follow_up messages, and the Announce message is used to establish the synchronization hierarchy.

A. PTP State Machine

PTP state machine is mainly responsible for detecting initialization state, fault state in abnormal situation, listening state in normal situation, pre_master clock state, master clock state and slave clock state when the device or system power on. State transitions is as shown in Fig.2

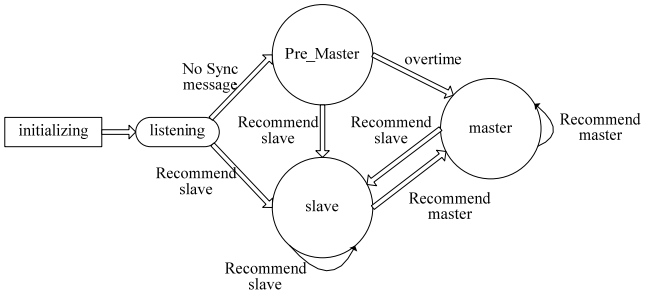

Figure 2. PTP state machine topologies

B. PTP Clock Synchronization

IEEE 1588 precision clock synchronization of the executive need two order section [1]:

- BUILD SYNCHRONIZATION SYSTEM: through the best master clock algorithm, select the master clock in the synchronization system, then build a master-slave synchronization system.
- SYNCHRONIZATION PROCESS: Through message exchange between master and slave, calculate the time difference, and synchronize the local clock in order to maintain the same absolute system time.

1) Build synchronization system

PTP network system can be composed by one or more PTP sub-domains, each sub-domain include one or more clock communicated with each other, and each clock could be in the following two states: master clock state and slave clock state, but master clock is only one. Clock state is determined by BMC (Best Master Clock algorithm). BMC requires that each device in the PTP network system provides a data set (DS), the nature, quality, stability, unique identifier and preferred setup of its local clock are described by the DS [2]. Each piece of equipment added to the PTP network system, will broadcast its clock DS, and accept other equipment DS in the system through the comparison between the DS to determine its present state and future state. The best master clock algorithm consists of two parts:

a) DS comparison algorithm

Clock DS contains the relevant parameters to describe the clock feature and the clock variance is an important parameter of said clock performance. In the PTP protocol, the clock variance that represents the clock

performance is calculated based on the Allan variance [3], Allan variance can be defined as below:

$$\sigma_\gamma(\tau)=\left[\frac{1}{2(N-2)\tau^2}\times\sum_{k=1}^{N-2}(x_{k+2}-2x_{k+1}+x_k)^2\right]^{1/2} \quad (1)$$

In formula (1), x_k, x_{k+1}, x_{k+2} are the time residual of time t_k, $t_k+\tau$ and $t_k+2\tau$, τ is a period of sampling in measurement. But Allan variance is given the statistical characteristics of the time based on oscillator frequency change, and the clock variance is the time difference statistic which measured based on relative reference clock. Therefore, so the clock variance calculation should be changed accordingly on the basis of the Allan variance. Taking the PTP variance (clock variance) $\sigma_{PTP}^2=\tau^2\times\sigma_\gamma^2/3$ into equation (1), it can get changed clock variance formula:

$$\sigma_{PTP}^2=\frac{1}{3}\left[\frac{1}{2(N-2)}\times\sum_{k=1}^{N-2}(x_{k+2}-2x_{k+1}+x_k)^2\right] \quad (2)$$

In which x_k, x_{k+1}, x_{k+2} is the clock residual between the measured time and the local reference clock of time t_k, $t_k+\tau$ and $t_k+2\tau$, τ is a period of sampling in PTP variance. About sampling period must comply with synchronous interval [4]. The clock DS stored in each clock as well as send and receive synchronization message. Confirmed the master clock by comparing the clock value of the variable in DS A and DS B, the workflow is as shown in Fig. 3.4.

The DS comparison algorithm is based on the comparison between two sets of data. These data can be described with following properties:

- Priority 1: the choice of the master clock is based on the user settings.
- Class: define the clock ATI traceability.
- Accuracy: define clock accuracy.
- Variance: define clock stability.
- Priority 2: define user-defined clock sequence when appear equivalent clock.

The DS comparison algorithm flow is roughly divided into two parts:

- DIFFERENT GRANDMASTER(GM) CLOCK: select the better clock as master clock for PTP sub-domain based on comparing the each property of grandmaster clock rather than the properties of local clock, which will help to maintain system stability. Shown in Fig. 3.
- SAME GRANDMASTER(GM) CLOCK: according to the distance of the local clock and the master clock or compare the ports number between receive and send to distinguish the merits of the clock, select the appropriate clock as the master clock. Shown in Fig. 4.

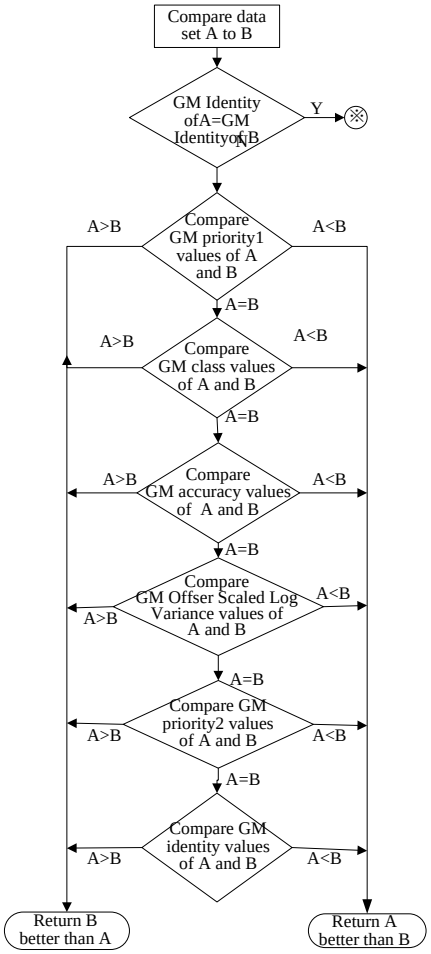

Figure 3. Data set comparison algorithm(1)

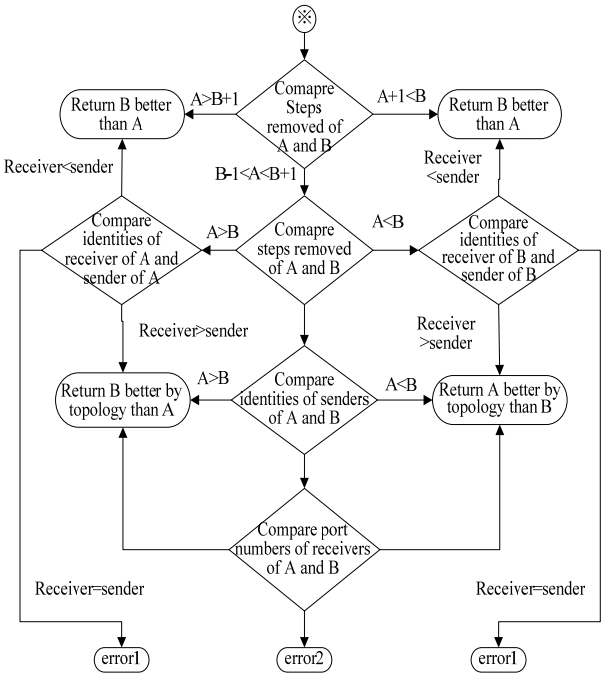

Figure 4. Data set comparison algorithm(2)

Parts of the DS comparison algorithm program are as follows:

```
if(memcmp(msgA->grandmasterId,msgB->
   grandmasterId,sizeof(ClockIdentity)) == 0)
{
   if (tmpA > (tmpB + 1))
   {
        ptp_clk_id(msgB-
>hdr.src_port_id.clock_identity);
         return 2;
   }
else if ((tmpA + 1) < tmpB)
      {
        ptp_clk_id(msgB-
>hdr.src_port_id.clock_identity);
         return 0;
      }
      else if (tmpA > tmpB)
      {
         if(portA)
         {
            ret=compare_clock_id(portA->
                clock_identity,msgA                    ->
                hdr.src_port_id.clock_identity);
            if(ret==-1)
            {
                ptp_clk_id(msgB                        ->
                hdr.src_port_id.clock_identity));
                return 2;
            } else if (ret == 1)
                {
                        ptp_clk_id(msgB->
                        hdr.src_port_id.clock_identity)
                        );
return3;
}
else
{
        return -1;
}
}
 else
{
        ptp_clk_id(msgB-
>hdr.src_port_id.clock_identity));
        return3;
 }
}
else if(tmpA<tmpB)
{
       if(portB)
       {
          ret=compare_clock_id(portB->clock_identity,
          msgB-> hdr.src_port_id.clock_identity);
          if(ret==-1)
          {
                ptp_clk_id(msgA->
                hdr.src_port_id.clock_identity));
                return 0;
          }
         else if (ret == 1)
         {
                ptp_clk_id(msgA->
                hdr.src_port_id.clock_identity));
                return1;
         }
         else
         {
                return -1;
         }
 }
else
{
        ptp_clk_id(msgA-
>hdr.src_port_id.clock_identity));
        return 1;
}
}
```

b) State decision algorithm

The local clock port state is depended on state decision algorithm by multiple operating DS comparison algorithm, the core content is the DS comparison algorithm. First using data set comparison algorithm compare that whether the received synchronization message is better than the local clock default DS, then set local clock as slave clock if it is , or as master clock if not. In the IEEE 1588 standard, the grade of the clock is divided into 0, 1, 2, 3, 4 and 255, a total of six kinds. The smaller the value, the higher level, the stronger property, which grades 1, 2, are recommended as the standard reference clock source, 255 clock is only as a slave clock. The workflow is as shown in Fig. 5.

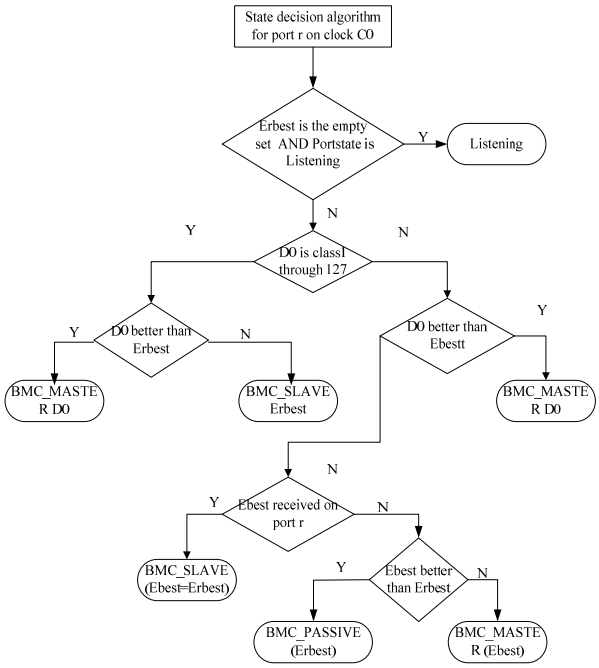

Figure 5. state decision algorithm

In Fig. 5, state decision algorithm judges the state of the port r on the C_0 at first, then judges the clock grade of the DS D_0 on the port r. According to the clock grade, compare with D_0 and the relationship between Erbest and Ebest by calling the DS comparison algorithm, in order to judge the state of each clock port.

Parts of the state decision algorithm program are as follows:

```
if ((foreign_elem_p[index].data_p==NULL)&&(port->
port_dataset.port_state==PORT_LISTENING)&&(!por
t  -> announce_recv_timer_expired))
{
    continue;
}
if ((ptp_ctx ->
parent_dataset.grandmaster_clock_quality.clock_class
>= 1 )&& (ptp_ctx ->
parent_dataset.grandmaster_clock_quality.clock_class
<= 127)&&(ptp_ctx -> clock_state ==
PTP_STATE_LOCAL_MASTER_CLOCK))
{
   if (foreign_elem_p[index].data_p)
   {
      ret =AnnounceDataComparison(D0,NULL,
      &foreign_elem_p[index]. data_p -> msg,
      &foreign_elem_p[index].data_p -> dst_port_id);
    }
   else
   {
       ret = 0;
   }
   if (ret < 0)
   {
       continue;
   }
  if((ret==0)||(ret==1))
  {
       ptp_bmc_update(ptp_ctx,port,BMC_MASTER_
       M1,NULL);
  }
  else
  {
       ptp_bmc_update(ptp_ctx,port,
       BMC_PASSIVE_P1,
       foreign_elem_p[index].data_p);
  }
}
```

2) synchronization process

IEEE1588 precision clock synchronization protocol is defined 4 message types: Synchronization message, Follow_up message, Delay request message, Delay request response message [5]. PTP network system through the exchange of four kinds of messages, calculates the time offset and delay between slave clock and master clock. Slave clock adjust its time based on the offset and delay, so as to achieve the time synchronization with the main clock. The whole process is divided into two stages: offset measurement and delay measurement.

First stage: Fixed the time offset between master-slave clocks, is called offset measurement. In this stage, Master clock in accordance with set time intervals (typically 2 seconds) periodic sends clock synchronization message to slave clock by multicast way. The message contains the estimated time of messages sent, slave clock records the received synchronization message time TS1. After a certain time delay, the master clock continue issued follow up messages by multicast way, the message contains the exact time TM1 of synchronization message issued, and slave clock calculated offset based on the information of the sync message and the following follow up message:

$$offset = TS_1 - TM_1 - delay \quad (3)$$

Second stage: network delay measurement. Slave clock sends delay request message to master clock by multicast way and records sent time TS2. Master clock records its arrival time TM2 when delay request massage arrival, and then master clock sends delay request response massage to the corresponding slave clock by multicast way. The massage contains the accurate time TM2, then figure out the delay after slave clock receive delay request response massage:

$$delay_2 = TM_2 + offset - TS_2 \quad (4)$$

In which assumed that the network delay is constant, this means that delay1=delay2, according to equation (3)and(4) can calculate that

$$offset = [(TS_1 - TM_1) + (TS_2 - TM_2)]/2 \quad (5)$$

$$delay = [(TS_1 - TM_1) - (TS_2 - TM_2)]/2 \quad (6)$$

Synchronous process is as shown in Fig. 6.

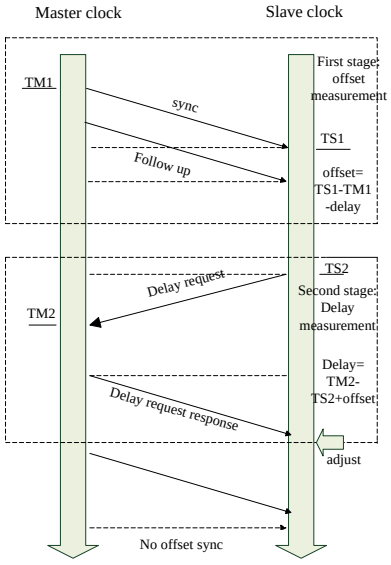

Figure 6. Clock synchronous processes

III. IEEE 1588 ACCURATE CLOCK SYNCHRONIZATION PROGRAM FLOW CHART

Program flow chart is as shown in Fig. 7.

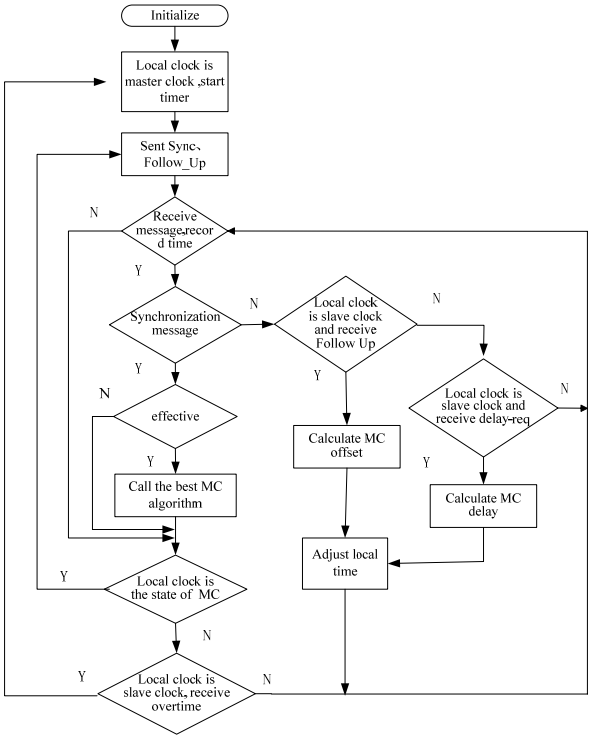

Figure 7. Program flow chart

After system power on, the DS of the clock port initialized at first, then set the local clock as master clock and enable the timer to record the local clock time. According to the clock DS to create synchronization message and send to PTP network system, the system will record the accurate time of the synchronous message sent, create follow up message contained accurate time, and the synchronization message will be issued after a certain time. At the same time of synchronization message and follow up message sent , the clock port is in listening state to monitor the external packet, once the message arrival ,received it immediately, including to record the receive time, to judge the type of message, if the received message was an valid sync message, then to call the best master clock module to determine the status of the local clock, if the local clock is running as a best master clock algorithm is still the main clock, it will send continually periodic sync message and follow up message into network systems in accordance with former set time interval. If the master clock receives a delay request message sent from slave clock, it will record delay request message arrival time, create a delay request response message with accurate time of reaching to the main clock and save into the message and sent to the slave clock. If the local clock becomes slave clock after running the best master clock algorithm, the system will create delay request message, send to the master clock and record the accurate send time, continue to wait to receive the follow message. When local clock receives a follow up message, it will take out the master clock issue sync message and accurate send time from the follow up message. When slave clock receives a delay request response message, it will remove out the accurate time that the delay request message reach to the master clock from the message. According to the value of the record, the local clock time would be adjusted by calculating the time delay and offset between the master clock and the slave clock. When the local clock is in the status of the slave clock, the system will start to receiving time out timer and to force set the local clock state to the main clock state and send message to the network if message is not received within the set time[6] [7].

The initialization program is shown as follows:

```
if (argc != 2)
{
        file = DEFAULT_CFG_FILE;
 }
 Else
{
        file = argv[1];
}
Memset (&ptp_cfg, 0, sizeof ( structptp_config ));
ret = read_initialization (file);
if (ret != PTP_ERR_OK) {return;}
memset (&ptp_ctx, 0, sizeof(struct ptp_ctx));
ptp_ctx.ports_list_head = 0;
init_default_dataset(&ptp_ctx.default_dataset);
ret=ptp_initialize_packet_if(&ptp_ctx.pkt_ctx);
if (ret != 0)
{
        return;
}
ret = initialize_os_if(&ptp_ctx.os_ctx);
 if (ret != 0)
 {
        return;
 }
ret=ptp_initialize_clock_if(&ptp_ctx.clk_ctx, file);
if (ret != 0)
{
        return;
}
init_current_dataset(&ptp_ctx.current_dataset);
init_parent_dataset(&ptp_ctx.parent_dataset);
init_time_dataset(&ptp_ctx.time_dataset);
init_sec_dataset(&ptp_ctx.sec_dataset);
```

IV. SYNCHRONIZATION MODEL AND TEST RESULT

A. Synchronization Model is As Show in Fig. 8.

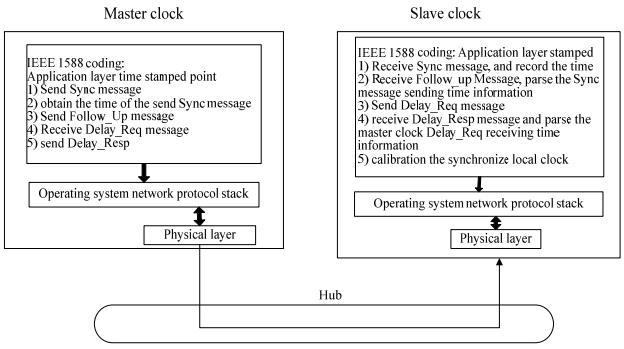

Figure 8. Master-Slave synchronization model

After determined the master-slave relationship according to the protocol, master clock mainly complete to send Sync message, record the exact time of synchronization message sent, and record Follow_up message sent, Delay_Req message received and time in Follow_up message, deposited the time to Delay_Resp message and sent. Slave clock main task is to receive Sync message and record received time, receive Follow_up message, parse out the exact time of the master clock to send Sync message, send Delay_Req message and record the time, send Delay_Resp message and parse out the exact time of the master clock to received Delay_Req message. According to the local record time and the parsed out master clock record time, adjust the local time to complete the synchronization with the master clock.

B. Test Result

Taking two embedded Linux system equipment which connected by the Hub as test object, run the PTP protocol program in application layer that can realize clock synchronization of the two machines. Operation interface under the Linux is as shown in Fig. 9 and Fig. 10.

Obtained from Fig. 8 of master clock operation

The current time of master clock is:

1329980384s, 854725000ns

Obtained from Figure 6 of slave clock correction interface

The current time of slave clock is:

1329980384s, 854685000ns

ΔDelay=854725000ns-854685000ns=40000ns=40μs

According to the calculated offset and the amount of the delay, the slave clock is corrected closed to the local clock, the accuracy can reach to 40us, basically reach the sync with the master clock.

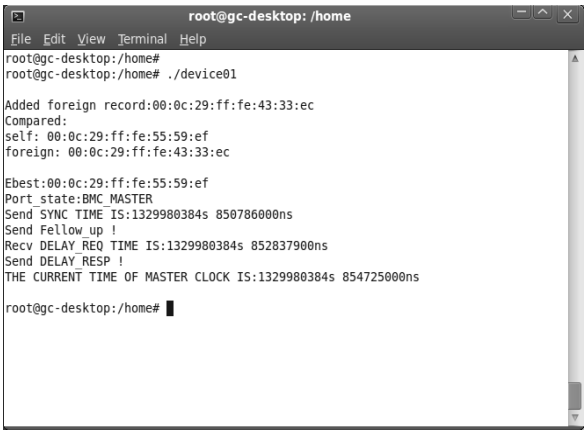

Figure 9. Master clock operation interfaces

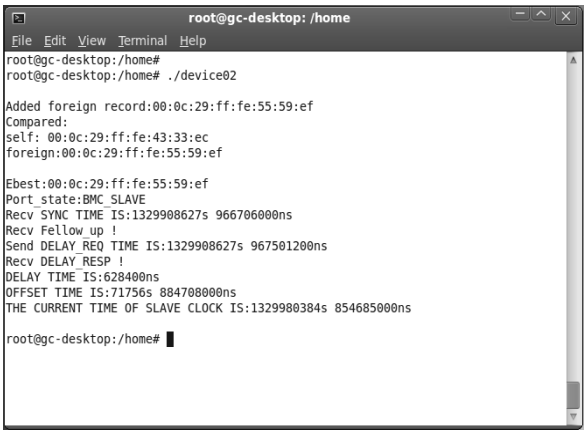

Figure 10. Slave clock fixed interface

V. CONCLUSION

PTP protocol is a precision clock sync protocol defined in the IEEE-1588, mainly for relatively localized, network system. Due to the advantages of the PTP protocol is simple, occupied fewer network and computing resources make it widely used in distributed systems, which is one of the key technology for distributed measurement and control, also is an important part of the new generation of the LXI measuring instrument. Clock synchronization precision for distributed network systems will be advanced greatly by using of IEEE 1588 to achieve Clock synchronization.

REFERENCES

[1] Yijun Liu, Zhenkun Li Design and program multi-processor platform for high-performance embedded processing JOURNAL OF SOFTWARE, VOL. 4, NO. 10, DECEMBER 2009 1069-1075

[2] Andreu Carminati, Rômulo Silva de Oliveira, Luís Fernando Friedrich. Implementation and Evaluation of the Synchronization Protocol Immediate Priority Ceiling in PREEMPT-RT Linux[J]. Journal of Software, Vol 7,No 3 (2012),516-525

[3] IEEE Std 1588–2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control System [S],July ,2008.
[4] Dai Baofeng, Cui Shaohui, Chang Jian. Implementation and analysis of best master clock algorithm of IEEE 1588[J], Instrumentation Technology, 2008(2):29–31.
[5] John Edison, The application of IEEE 1588 to test and measurement systems[R],AGILENT,2005:1-2
[6] Yu Zhilan, Li Zhi . best master clock algorithm of precision clock synchronization protocol[J].Electronic Automation Equipment, 2009,29(11):74–77.
[7] Liu Mingzhe. Implementation of clock synchronization algorithm by software method base on IEEE 1588 protocol [J].Chinese Journal of Scientific Instrument, 2006,27(6):2009-2011.
[8] Yu Zhilan, Li Zhi. Implementation and analysis of IEEE 1588 precision clock protocol[J]. Electronic Measurement Technology, 2009,32(4): 56-58.
[9] Song Jinbing, Sun Haibing. Linux Network Programming [M]. BeiJing : Tsinghua University Publishing House, 2010.

Weikang Qian He was born in Nov.1957.Received his Master degree from University of Shanghai for Science and Technology in 1992. He is currently a grade 5 Professor, Master mentor, former vice-director of Department of Electrical Engineering in USST and advanced member of China High-tech Industrialization Association. His research interest covers electronics, FPGA embedded system and control engineering & application. Corresponding author of this paper.
E-mail: elelsec@usst.edu.cn

Chao Guo He was born in 1985. He is now a postgraduate student of University of Shanghai for Science and Technology under Prof.Qian, and currently internship in China Orient Institute of Noise and Vibration. His research interest covers pattern recognition and intelligent system, embedded system.
E-mail: guochaohappy123@163.com

Junhui Mei She was born in Oct.1987. She is now a postgraduate student of University of Shanghai for Science and Technology under Prof. Qian, her research interest covers FPGA application and embedded system.
E-mail:may.0714@163.com