The Design and Implementation of Time Synchronization for CNC Systems Based on Switched Ethernet

Haibo Yang, Hu Lin, Yaodong Tao, Jin Li
The National Engineering Research Centre for High-end CNC of China, Shenyang Institute of Computing Technology, CAS
College of Computing & Communication Engineering, Graduate University of the CAS
Shenyang, China
E-mail: yanghaibo@sict.ac.cn

***Abstract*—It requires accurate synchronization among the MCU, servo controllers and sensors in CNC systems for time stamping of data and motion control. This paper presents the design and implementation of time synchronization for CNC systems which employ switched Ethernet as its communication module. The method adopts the clock of switches as the master clock instead of that of MCU which usual is. To obtain accurate time synchronization, it uses a low-level time-stamping scheme and blocks the time synchronization frames until the actual time is equal to the time in the payload of the frames. A preemptive mechanism is introduced to make sure the time synchronization frames can be transmitted immediately when the actual time is equal to that. Finally, experimental results are shown and discussed.**

***Index Terms*—CNC Systems; Switched Ethernet; Preemptive Queues; Time Synchronization**

I. INTRODUCTION

The CNC technology is the foundation for realizing automation, flexibility, integration of the manufacturing industrial. In traditional CNC systems, it is often based on centralized control architecture and can't fit the requirement of higher speed, more precision and rapidly developing of machine controller adapted to new production systems (e.g. CIMS and CAD/CAM). So one important trend of the CNC technology is toward to digital and networking [1]. One may construct CNC network through many ways, such as serial, fieldbus and MAP technology. These methods can meet the data exchange needs of the CNC systems in certain aspects, but all the ways have flaws in applications such as bad compatibility, limited network bandwidth, high cost, and difficulty when integrating with CISM and so on. At the same time switched Ethernet has been considered as the most attractive solution for supporting real-time communications in industrial communication because it has many advantages such as large bandwidth, full-duplex and prioritized frame transmission. Unlike traditional shared Ethernet, switched Ethernet provides a private collision domain for each of its ports and promises to support real-time communication. Many works have been done to use the switched Ethernet into various networked control systems [2] [3] [4]. In their works the time synchronization is one of the most important focuses to study because in switched Ethernet there are switches between any two devices and the message delay through switches which is always time variant must be taken into account. It is known that the time synchronization of motor control in CNC systems must achieve 1us or below which is the most precise requirement in various networked control systems. This paper will present the time synchronization mechanism of CNC systems which are based on switched Ethernet. To obtain accurate time synchronization, we introduce the preemptive priority queuing in the design of the switch and use the clock of the switch as the time reference instead of that of MCU. This paper consists of five sections, including the introduction section. Section 2 describes the fundamental of the CNC systems based on switched Ethernet and the basic principles of time synchronization. Section 3 outlines time synchronization mechanism we present in detail. Section 4 presents the test result. Finally Section 5 concludes the paper.

978-1-4244-5848-6/10/$26.00 ©2010 IEEE

II. STATE OF THE ART

The CNC systems based on switched Ethernet can be shown in Figure 1.

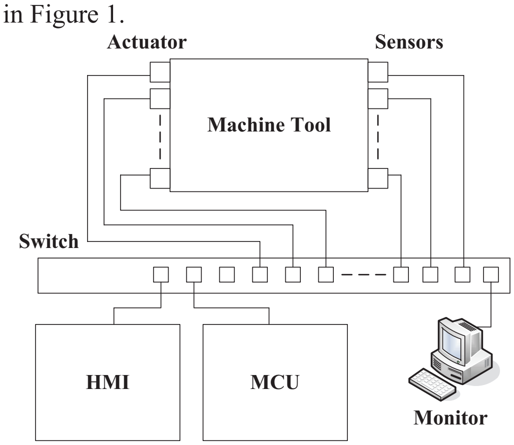

Figure 1. The Architecture of the CNC system based on switched Ethernet

By connecting MCU, HMI, servo controllers, sensors, PCL and I/O with an Ethernet switch, it forms a distributed control system with star topology. The star topology has much advantage than ring or daisy chain topology because problems of ring or daisy chain scheme in use is that the bandwidth utilization efficiency and delay bound of real-time packages highly depends on every node's performance, especially there are some embedded systems with little computing resource, while the star topology separates the communication efficiency and the nodes' performances, and it is easy to replace, install or remove hosts or other devices.

This system can be seen as a MIMO complex networked control system with communication delay. IEC has provisionally defined five levels of synchronization accuracy (IEC 61850-5, section 12.6.6.1 and 12.6.6.2 [5]) for various systems from T1 to T5 varying from 1ms to 1us. Because in the CNC systems there are motor control tasks, the level of synchronization accuracy is T5, i.e., 1us.

Many theory and methods have been proposed for synchronizing clocks in distributed systems in recent years. The NTP (Network Time Protocol) and its subset SNTP (Simple Network Time Protocol) proposed by Mills and IETF group [6] with a precision of some milliseconds are important ones as one public domain synchronization method. To get more precise, in [7] Mills describe engineered refinements in the form of modified drivers and Unix Kernel codes for improving the time stamping accuracy within NTP time server. By these modifications a timekeeping precision of few microseconds for an Ethernet network of workstations is reportedly practical. In 2002, IEEE released standard 1588 for time synchronization across small networks used in measurement and control [8]. IEEE 1588 is based on a protocol invented by Agilent. In short, IEEE 1588 is based on one node transmitting a time synchronization message containing the precise time of the previous time message. One implementation of IEEE 1588 achieved the accuracy of several hundred nanoseconds [9]. Although this protocol fulfills class T5 requirements on some types of local area networks, it is just as susceptible to random switch delays as the NTP. Another method is GPS (Ground Position System). However, GPS is not suited to be installed on many nodes in a network because of it requires an antenna. There are also many proprietary and patented solutions to achieve highly accurate time synchronization over Ethernet. ABB holds several patents [10] based on the concept of a time master broadcasting a tick and directly afterwards broadcasting a message of time synchronization stating when the tick left the time master. This two-phase method effectively removes the real-time OS scheduling unpredictability and the network access unpredictability. U.S. Philips holds a patent [11] on the usage of hardware-based Ethernet time synchronization and clock adjustments being able to precise time tag Ethernet packets. Hewlett-Packard has patented time synchronization scheme which is reported able to perform synchronization accuracy in the range of 200ns; but this is only over a single repeater hub network.

III. PROPOSED SYNCHRONIZATION METHOD

A. Synchronization Problems

Consider a time-reference clock and time-client clock. Let T denote the time measured at the time-reference clock, and let t denote the time measured at the time-client clock. The general synchronization problem involves determining constants α and β such that $T = (1 + \alpha)\ t + \beta$ within a specified accuracy and for a specified time interval. The α can be seen as the time-client clock frequency offset which is the deviation between the time-client crystal oscillator and the time-reference oscillator, and β is the absolute time offset. For crystal accuracy, we have $|\alpha| < 10^{-4}$. If there is a switch inserted between the time reference and the time client, a time variant variable is involved.

$$T = (1 + \alpha)\ t + \beta + p(t) \qquad (1)$$

Where $p(t)$ is a random delay function including the switching delay, queue delay and so on. Without the $p(t)$ α and β can be calculated by statistical, filtering and erasure methods, but with $p(t)$ it is difficult to do so.

B. Time Synchronization Model

Traditional CNC systems based on fieldbus usually synchronize serve controllers, PLC and other devices with the MCU. But in the CNC system, there are a switch between such devices and the MCU, so if we also use the MCU as the time reference, it will really hard to achieve T5 requirement because the delay $p(t)$ in the switch is time variant. So we proposed synchronizes the MCU and other nodes with the switch, because the switch is also an intelligentized system with clock and its position is the center in the star topology as shown in figure 2. More important is that every node has an exclusive full-duplex link with the switch and the data exchange between the node and switch will not suffer from collisions at all.

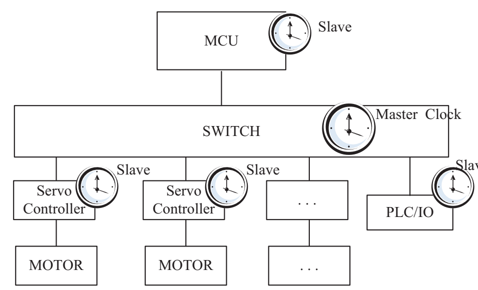

Figure 2. Time Synchronization based on the switch clock

The proposed synchronization method between the switch and other devices in the CNC system synchronizes the slave clock to the master clock by periodically exchanging synchronization frames like SNTP. As we known the SNTP can't achieve class T5 requirements, so some enhancements must be performed. Firstly, let's analyze the time stamp and its delay.

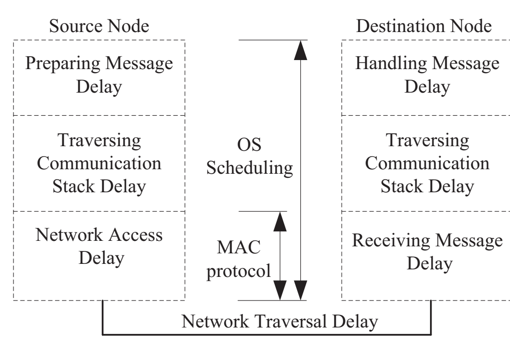

Figure 3 Time Synchronization Frame Delay

The delay from the time stamping of a time synchronization message of SNTP in switch until it is time stamped in the message destination device can be shown as figure 3.

Every time client and the time reference are connected directly, so the network traversal delay can be treated as a constant which is decided by the length of the network cable. Variations in the delays are due to OS scheduling unpredictability and network access unpredictability. To reduce the variations in the delay and to attain sufficient time-stamping accuracy, a low-level time-stamping scheme must be implemented both in the switch and other devices. There are three low-level time-stamping methods:

1) *Time stamping in (or close to) the Ethernet controller*

2) *Time stamping in the interrupt service routine(ISR) outside the RTOS*

3) *Time stamping in the Ethernet driver controlled by the RTOS*

To achieve class T5, we adopt the hardware time stamping in the Ethernet controller as shown in Figure 4.

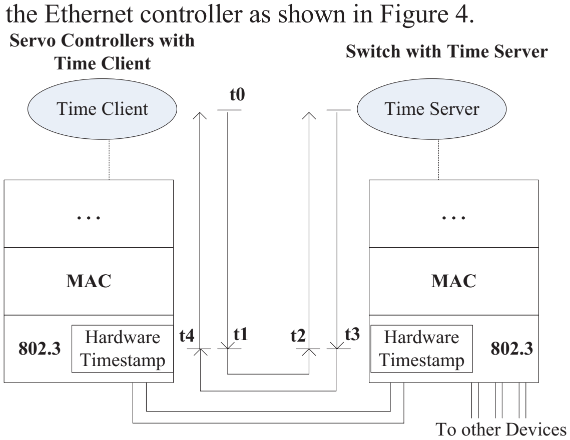

Figure 4 Time synchronization scheme

The process of time synchronization can be described below. The time client initiates a time synchronization request at time t0, the frame contains a time t1 where t1 > t0 and t1 is bigger enough than the maximum delay to transmit a message. The time synchronization request wouldn't be sent until actual time is equal to t1 given in the request frame payload. As soon as the time request frame enters the switch, it is time stamped in the hardware named t2. Then the time server reply the request with a time t3 when the reply will be transmitted, and the soon as the node with time client receives the reply packet it stamps the packet in the hardware we assume the time is t4. So we now have:

$$t2 = t1 + td + \Delta \quad (1)$$
$$t4 = t3 - \Delta + td \quad (2)$$

Where *td* is the one-way network traversal delay and Δ denotes the time offset between the time server and the time client. Because time client and the time server are connected directly via an exclusive link, the network traversal delay can be treated as a constant. Δ and *td* are then computed at the time client node using the following equations.

$$\Delta = (t2 - t4) + (t3 - t1) / 2 \quad (3)$$
$$td = (t2 + t4) - (t1 + t3) / 2 \quad (4)$$

Time is synchronized by advancing or retarding the slave clock to cancel Δ derived by eq.(5). Namely, the clock should be advanced for $\Delta > 0$ and retarded for $\Delta < 0$. In this process, the slave clock must be gradually adjusted to avoid time jumps.

C. Preemptive Priority Mechanism

To guarantee at t1 and t3, the time synchronization frame can be transmitted, a preemptive mechanism should be adopted to minimize the delay and delay variation. With preemptive priority queuing, if the system receives a high priority frame while processing a low priority frame, it stops processing the latter and begins processing the former. The system resumes processing the low priority frame after the high-priority frame is processed. In this scheme, we define the time synchronization frame with highest priority. To implement preemptive priority queuing, the receiver must identify and extract high priority frames besides reconstructing low priority frames as necessary. In [12], S. Yanagimachi implements such functions by inserting special codes before and after each high priority frame which is illustrated in Figure 5.

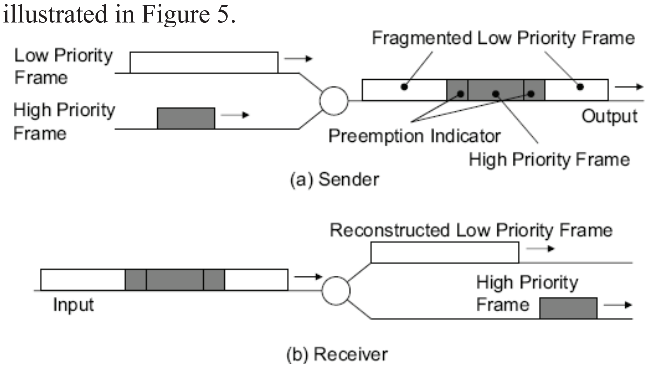

Figure 5 Preemptive Priority Control at the sender and receiver

We use a commercial Ethernet multiplexer from NTT Electronics [14] to implement preemptive priority mechanism in a method similar to that used in [12]. The difference between our implementation and that of [12] is that this study places preemptive priority queuing between the MAC and PHY layer while [12] places it in the PHY layers as shown in Figure 5. This implementation is expected to be more economical than that of Ref.[12].

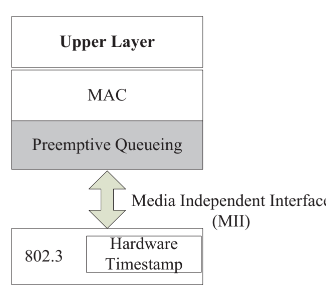

Figure 6 Layer diagram for Preemptive Priority Mechanism

When a time synchronization frame reaches the Preemptive Queue with a sending time t, it firstly be queued, and then the preemptive priority mechanism will interrupt the sending of other frames when the actual time is t. After the time synchronization frame is sent, the frame which is interrupted will continue to be transmitted.

IV. EXPERIMENT

To verify the performance of the proposed scheme, we developed a customized switch and test nodes with time client based on the ADSP-BF518 processor from ADI. The processor has a built-in Ethernet media-access controller (EMAC) module, and it can detect and provide hardware time stamps for all pre-defined event messages, including both incoming and outgoing packets by its TSYNC module which keeps monitoring the hardware interface between the MAC controller and the Ethernet physical interface transceiver, that is, the media independent interface (MII). The detection of event messages is designed to be programmable. A complete proposed time synchronization implementation was built on ADSP-BF518 processor as shown in Figure 7.

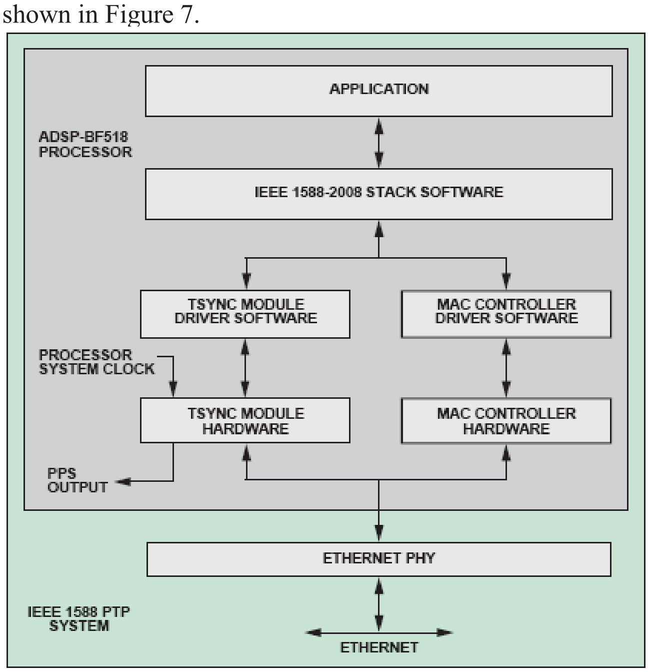

Figure 7 Implementation of Time Synchronization Based on ADSP-BF518

The TSYNC module of the processor detects incoming and outgoing time synchronization messages and use hardware to time-stamp event messages and uses the MAC controller driver to implement the preemptive priority queue. The Ethernet PHY is National Semiconductor DP83848, chosen because of its low jitter delay characteristics. For simplicity, the processor's system clock (80 MHz) was chosen to be the TSYNC module clock source.

The measurements were made as illustrated in Figure 8. The two clocks were connected directly via class 5 cable. Each clock has a 1-PPS test point. An oscilloscope(TDC-GP2) was used to measure the relative offsets between the master and the slave.

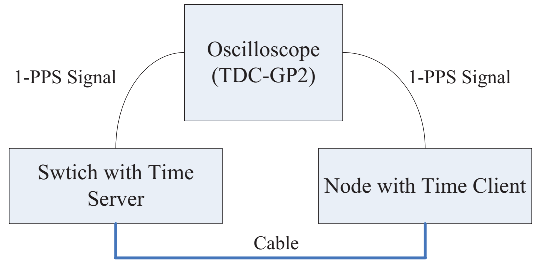

Figure 8. The measurement architecture of the time synchronization

Because the switch and the node with time client operate at 80MHz, it offers a 12.5ns sampling granularity. It means incoming and outgoing frame at the master and slave introduce a 12.5ns uncertainty or jitter. Hence a minimum of 25ns jitter and offset deviation can be expected from the setup. Table I summarized the measured performances at different sync frame transmit intervals.

TABLE I. EXPERIMENT RESULTS

Synchronization Interval(s)	Offset Average(ns)	Minimum offset(ns)	***Maximum offset(ns)***	***Jitter(ns)***
1	-0.8	-103	+96	199
2	-0.9	-231	+198	429
3	-0.6	-435	+402	837
4	-0.8	-640	+617	1257
5	-0.7	-912	+944	1856

The results show that the slave synchronizes in the range of 1.5ns to the master in average; however the jitter numbers are still showing variations in the range of 200ns when using a sync frame interval of 1 second. From the result, it shows that the synchronization interval has an important influence on the precision. When we choose the synchronization interval below 3 second, it can achieve the class T5 requirements used in the CNC systems.

V. SUMMARY AND CONCLUSIONS

In this paper we have presented a time synchronization method for CNC systems based on switched Ethernet. This method adopts a low-level time-stamping scheme and queue the time synchronization frame until the actual time is equal to the time in the payload of the frame. A preemptive mechanism is introduced to make sure he time synchronization frame can be transmitted immediately when the actual time is equal to the time in the payload. Then we give the implementation of time synchronization for measurements based on ADSP-BF518 processor and the experiment results. The results show that the proposed method accurate enough to CNC systems.

ACKNOWLEDGMENT

This study was supported by the grants from Major Projects of. National Science and Technology of China with No. 2009ZX04009-022. This work has been performed under the National Engineering Research Centre for High-end CNC of China. Consortium members are acknowledged for their useful comments.

REFERENCES

[1] Dong Yu *et al*, “An Open CNC System Based on Component Technology”, IEEE Transactions on Automation and Engineering, Vol 6, No.2, pp302-310, April 2009

[2] J. Jaspemeit & P. Neumann, "Switched ethernet for factory communication", In Proc. of ETPA-8th IEEE Int'l Conf. on Emerging Technologies and Factory Automation,Antibes, France,2001

[3] J. Loser and H. Hartig, Low-latency hard real-time communication over switched Ethernet, ECRTS 2004, pp. 13-22, Jul. 2004.

[4] Dao Manh Cuong, Myung Kyun Kim and Hee Chan Lee, “Supporting hard real-time communication of periodic messages over switched ethernet”, In Proceedings of IFOST2006, IEEE Press,pp.419-422, OCT. 2006

[5] IEC 61850 Communication Networks and Systems in Substations, Part 5: Communication Requirements for Functions and Device Models CDV Feb. 2001, Part 7-2: Basic Communication Structure for Substations and Feeder Equipment - Abstract Communication Service Interface (ACSI), CDV March 2001

[6] D.L. Mills, “In ternet time synchronization: the Network Time Protocol”, IEEE trans. Communications COM-39, 1482-1493,1992

[7] D.L. Mills, “Precision synchronization of comuter network clocks”, ACM Computer Communication Review 24, 28-42,1993

[8] IEEE, “1588 IEEE Standard for a Precision Clock Synchronization Protocal for Networked Measurement and Control Systems”, Draft May 2007 (Version 2).

[9] T. Cooklev, J. C. Eidson, and A. Pakdaman, “An implementation of IEEE 1588 over IEEE 802.11b for synchronization of wireless local area network nodes,” IEEE Trans. Instrum. Meas., vol. 56, no. 5, pp. 1632–1639, Oct. 2007.

[10] U.S Patent, No307728:Method for Providing Time Synchronization in a Netwrok, No307768:Enhanced Accuracy Time Synchronization, No307769: Ethernet Time Synchronization

[11] U.S Patent 4,815,110: Method and a system for synchronizing clocks in a bus type local network.

[12] S. Yanagimachi, T. Yoshikawa, T. Baba and K. Fukuchi, “An architecture of CIAO (Cut-In-Any Occasion) which makes a virtual private channel in a PHY layer of the Ethernet,” Technical Report of IEICE, CS2003-105, pp. 37–41, Nov. 2003.

[13] NTT Electronics, http://www.nel.co.jp/product/optical/gigabee2.html.

[14] Daniel Kohler, “A Practical Implementation of an IEEE 1588 supporting Ethernet Switch”, In Proc. of Precision Clcok Synchronization for Measurement, Control and Communications, IEEE Symposium on (ISPCS2007),pp. 134-137, Oct. 2007

[15] T. Skeie, S. Johannessen and O. Holmeide, “Highly accurate time synchronization over swtiched Ethernet”, in Proc. 8th IEEE Conf. Emerging Tech. Factory Automation, Oct,2001,pp. 195-204