ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

# Time Synchronization of Distributed Systems based on IEEE1588

Xiaohan Wei[1, a, *], Kaixing Cheng[1, b]

[1]Artificial Intelligence Key Laboratory of Sichuan Province, Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin, China

[a]weixiaohans@163.com, [b]1605221321@qq.com

## Abstract

**Because of the development of distributed systems, the accuracy of clock synchronization has become a key factor affecting the performance of distributed systems. The IEEE1588 precise clock synchronization protocol provides a synchronization technology based on network data packets. According to the content of the protocol, this paper designs the message sending and receiving process at the application layer. In addition, depending on the Linux kernel, the slave clock time is adjusted. The results show that the entire clock synchronization system can be applied to distributed systems that have certain requirements on clock accuracy.**

## Keywords

**Distributed systems; Clock synchronization; Packet Transport Network.**

## 1. INTRODUCTION

The distributed system has the advantages of high flexibility, remote controllability, and relatively low synchronization cost [1-2]. Therefore, it has a wide range of applications in the financial, military, and communications fields. In a distributed system, it is very important that each node has the correct clock information. For measurement and control and communication equipment, the requirements for synchronization accuracy often reach the microsecond or even nanosecond level [3-4]. In past applications, the system is often a non-distributed structure. Each terminal uses GPS or Beidou satellite as the clock source signal, and the result can reach the nanosecond level. But if GPS is used as the clock source, there are three disadvantages. One is that GPS is the global positioning system of the United States and needs to be considered in terms of safety. Second, because GPS is used as the clock source, additional acquisition modules need to be built, which increases the cost. The third is that in such as underground parking lots or large buildings, the GPS signal is weak and cannot achieve a good synchronization effect [5]. In view of the above considerations, IEEE1588PTP based on the Ethernet packet synchronization technology can solve the above problems well. And the PTP with hardware assistance can reach the error of nanosecond, and the pure software timestamping method can also reach the microsecond, which can satisfy most distributed synchronization systems [6].

## 2. DISTRIBUTED SYSTEM MODEL AND SYNCHRONIZATION MECHANISM

IEEE1588 defines a message-based time synchronization technology. The main models include ordinary clocks, boundary clocks, and transparent transmission clocks. The clock model, an ordinary clock can be divided into master clock and slave clock according to its position in the clock domain. The PTP system is a distributed system. It has a strict hierarchical structure. The master-slave clock nodes in the sub-clock domain of each level only synchronize to the local clock domain, and the synchronization message does not cross to other clock domains [7].

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

The PTP protocol is a distributed protocol, and its application structure is shown in Figure 1, which describes such a synchronization process: Firstly, these clocks form a master-slave hierarchical synchronization structure. The clock at the top of the hierarchy is called the best master clock [8]. Generally, an atomic clock or a rubidium clock is used to achieve the most precise timing. It is the most standard reference time of the entire synchronization system; secondly, the slave clock continuously exchanges PTP synchronization packets with the master clock in their own clock domain. When the slave clock completely receives all the packets required for one synchronization, it will calculate the offset between master clock and slave clock, and then adjust the local time to synchronize with the master clock; for the boundary clock, it can be split into multiple master clock models and a slave clock model to connect two different clock domains; the transparent transmission clock can be regarded as a transparent connection, which does not increase the transmission delay caused by the data link.

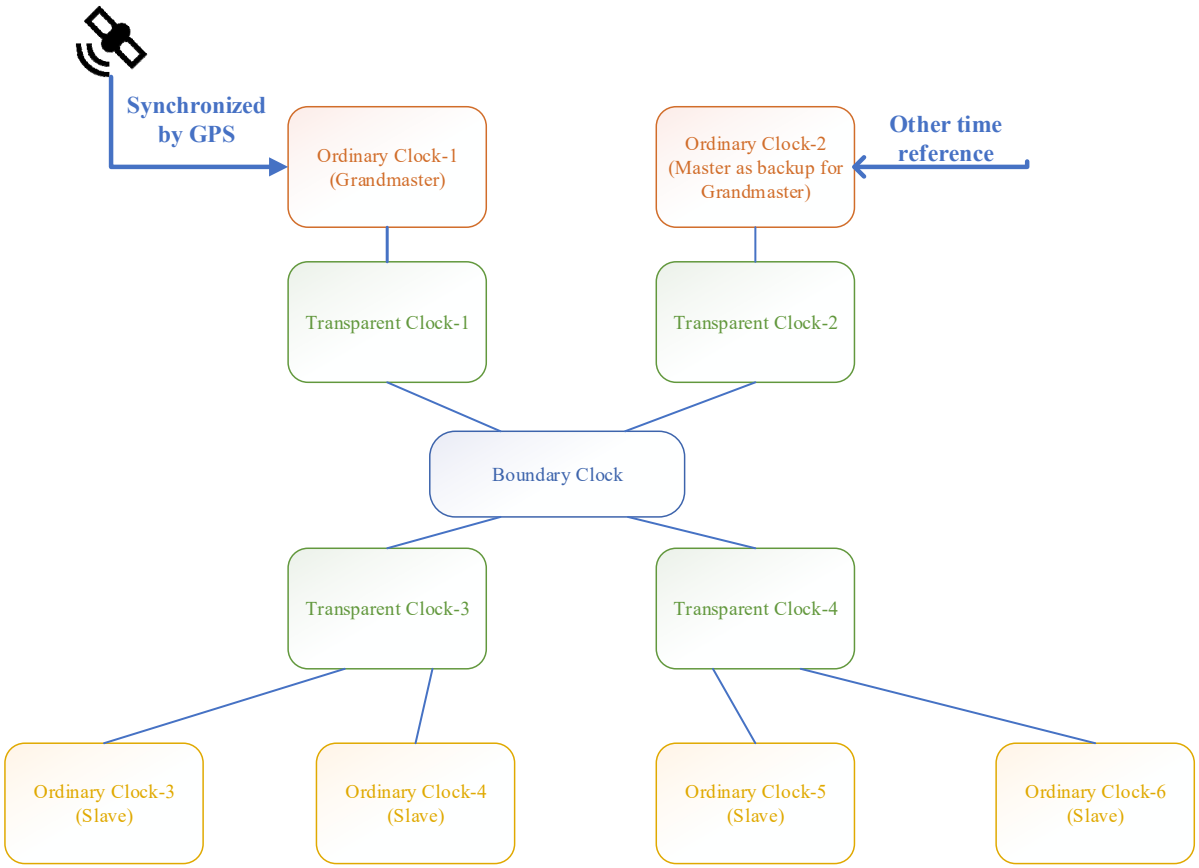


**Figure 1.** PTP typical application block diagram

IEEE1588 defines a message-based time synchronization technology. The main models include ordinary clocks, boundary clocks, and transparent transmission clocks. The clock model, an ordinary clock can be divided into master clock and slave clock according to its position in the clock domain. The PTP system is a distributed system. It has a strict hierarchical structure. The master-slave clock nodes in the sub-clock domain of each level only synchronize to the local clock domain, and the synchronization message does not cross to other clock domains [9].

The synchronization principle of IEEE1588 is to first use the BMC (Best Master Clock) algorithm to select the clock with the best performance as the grandmother clock. Secondly, according to the network structure, divide different clock domains. After this, clock synchronization starts, and the master and slave clocks exchange messages periodically. The slave clock uses the acquired information to adjust the local clock to achieve the purpose of clock synchronization, as shown in Figure 2. The specific message exchange steps are as follows:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

The master clock first sends a Sync message, and the timestamp t1 is stamped on the master, and then the t1 information is placed in the Follow_Up message and sent to the slave clock. When the sync message arrives at the slave, the timestamp t2 is stamped from the clock, and then the timestamp t1 is obtained when the Follow_Up message arrives. At this time, the slave clock needs to send a delay_request message to the master clock, and at the same time record the timestamp t3 of the delay request message sent by the slave clock. The master clock receives a delay request message and sends a delay_response message to the slave clock. Delay_response message, which contains the timestamp t4 of the master clock. According to the obtained timestamps t1, t2, t3, and t4, the transmission delay $D_{UL}$ between the master clock and the slave clock and the transmission delay $D_{DL}$ between the slave clock and the master clock can be shown in the equation below. The time between the master and slave clocks can be calculated. The calculation method is as follows:

$$D_{DL} + Offset = t_2 - t_1 \tag{1}$$

$$Offset - D_{UL} = t_3 - t_4 \tag{2}$$

$$Offset = \frac{t_2 - t_1 + t_3 - t_4}{2} + \frac{D_{UL} - D_{DL}}{2} \tag{3}$$

IEEE officials generally believe that the IEEE 1588v2 clock-synchronized communication link is symmetrical, that is, $D_{UL} = D_{DL}$. Then the calculation method of time offset will become in the equation(4).

$$Offset = \frac{t_2 - t_1 + t_3 - t_4}{2} \tag{4}$$

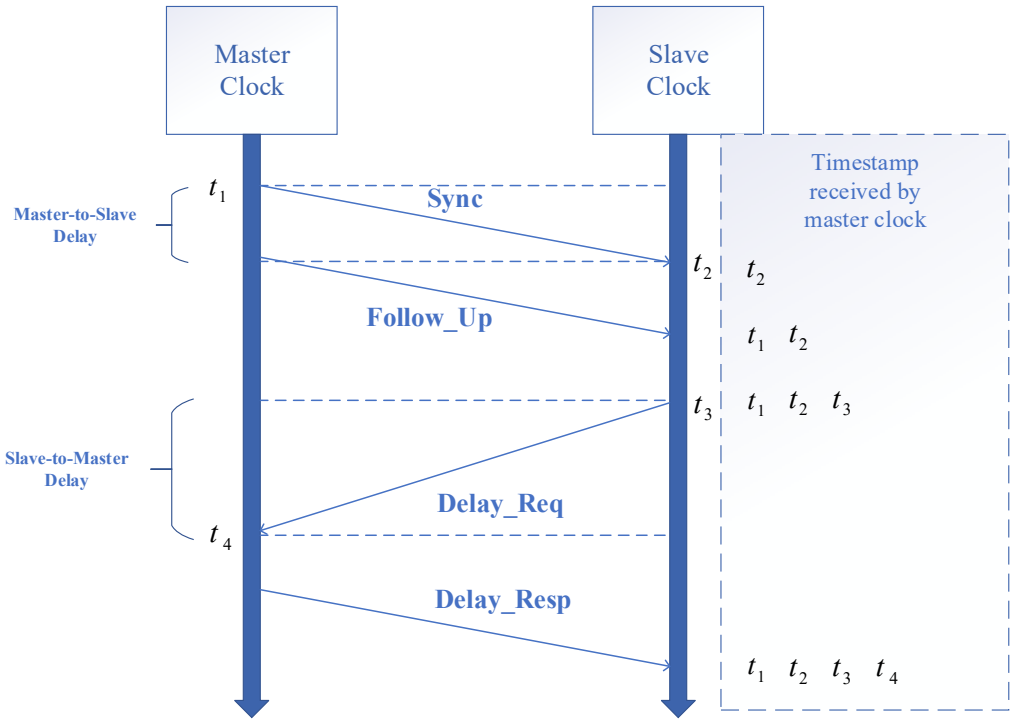


**Figure 2.** Principle of IEEE1588 clock synchronization

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

# 3. LINUX KERNEL AND MESSAGE SENDING AND RECEIVING PROCESS

## 3.1. Linux Kernel

Implementation of PTP based on Linux system requires the support of Linux kernel. The Linux kernel has integrated support for PTP after 3.1. Some socket options are defined in the Linux kernel to read or mark the timestamps. Secondly, a new interface is defined for the user plane and the kernel. The time stamp information of the slave clock and the master clock can be transferred to the user plane, so that the system clock of the device and the hardware clock of the device can interact. When the clock synchronization system starts to operate, the algorithm of clock synchronization is run on the user side to realize PTP content. Then the clock deviation is transmitted to the kernel through the interface between the user plane and the kernel. Finally, the kernel uses the driver to synchronize the external hardware clock, so as to achieve the synchronization of the master and slave clocks. In summary, the CPU implements the calculation of the protocol, the encapsulation of messages, and the transmission of time stamps in the system. Then use the adjustment information calculated by the user interface to correct the output of the local oscillator. The frame is shown in Figure 3. In the user plane, the protocol engine is mainly divided into two aspects. Execute the BMC algorithm before synchronization starts to select the optimal clock. In the data interaction stage, the specific adjustment value is calculated by the obtained timestamps. In the Linux kernel, the unit of low-precision timer is second, and the unit of high-precision timer is nanosecond. The displayed value is a specific number, which follows the operating rules of the Linux clock.

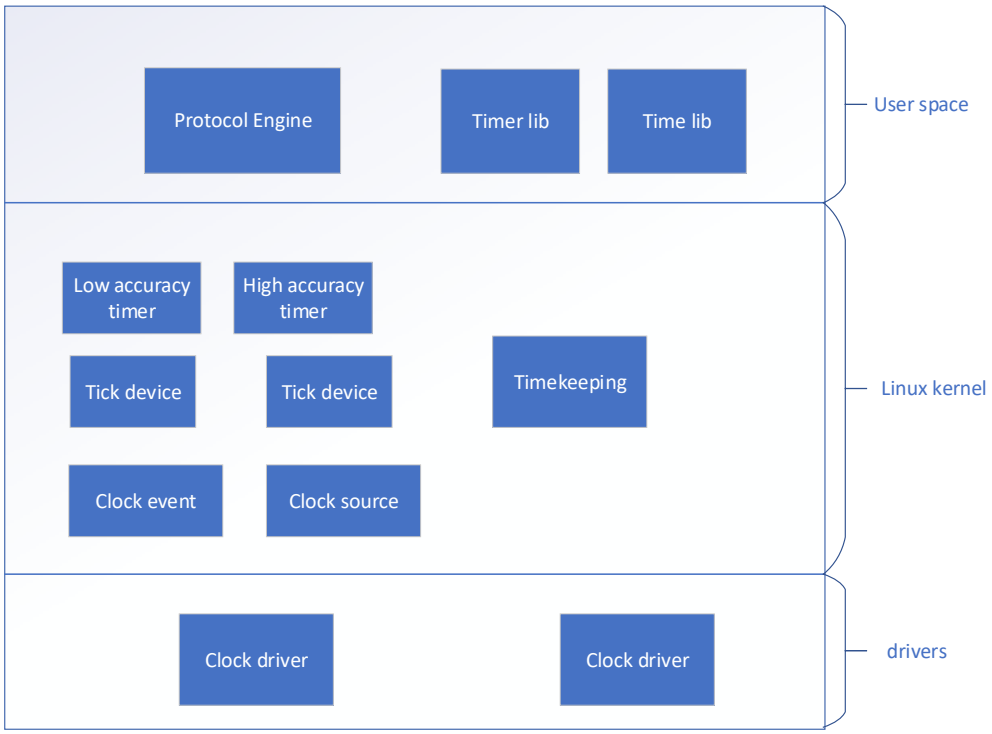


**Figure 3.** Software framework

## 3.2. Message Receiving Process

The message delivery method used in this project is shown in Figure 1, which contains four kinds of messages. They are Sync message, Follow_up message, Delay_Req message and Delay_Resp message.

At the beginning of synchronization, the master clock broadcasts the message, and the slave clock receives the message. Firstly, check whether the length of the message is greater than 0,

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

secondly check whether the version number of the message is true, and finally check whether the length of the message is greater than the length of the message header, so as to judge the legitimacy of the message. If all the above judgments are passed, it is judged whether the message is one of the above four kinds of messages. After that, different message types have different processing procedures. Figure 4 shows the overall flow of message reception.

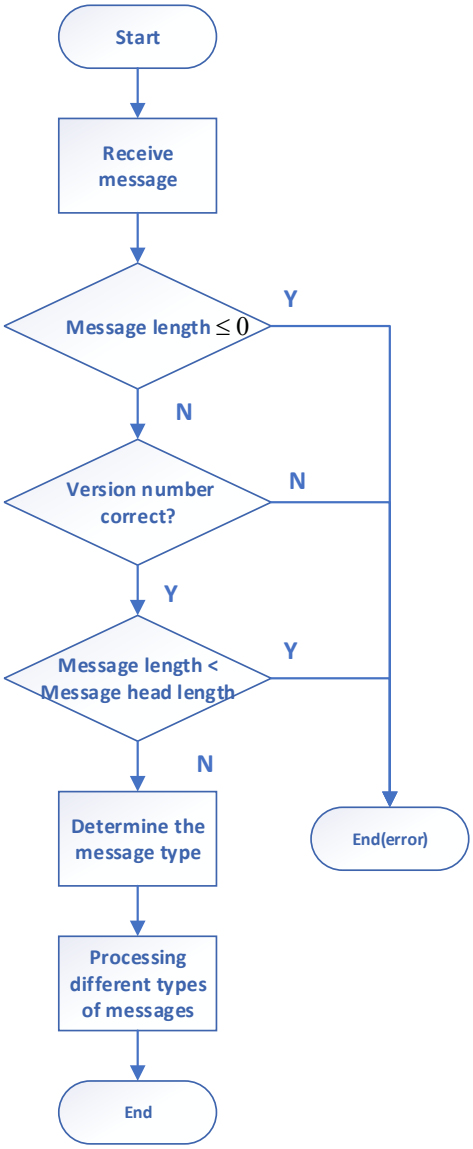


**Figure 4.** Message receiving

For the following process, no matter what message is received, the port status must be judged first. When the message is Sync, Follow_up or Delay_Resp, must first check whether the receiving end is a slave, and at the same time check whether the message is the master clock corresponding to the slave clock. Similarly, when the message is Delay_Req, check whether the receiving end is the master.

For Sync messages, there are two functions. One is to obtain the time stamp information when it arrives at the slave. The second is to determine whether to use the Follow_up message. If used, set the value of the Follow_up bit to 1. For Follow_up messages, there are also two functions. First, it carries the time stamp information sent by the Sync message. Second, when receiving the message from the clock, it must check whether the Follow_up message is a follow-up message of the previous Sync message. Delay_Req message and Delay_Resp message are the key

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

means to obtain the remaining two timestamps. It is worth noting that when sending a Delay_Resp message, it is first necessary to check whether the sending ID of the message is equal to the receiving ID of the Delay_Req message. Because in a distributed clock synchronization system, it is not only one-to-one clock synchronization. The sending method of the master clock message also adopts the broadcasting method. This means that it may be a one-to-many synchronization method.

## 4. TEST RESULTS

Set up the above two systems and set the synchronization time to once every 1 second. In an environment where two terminals are directly connected, the test results are shown in Figure 5.

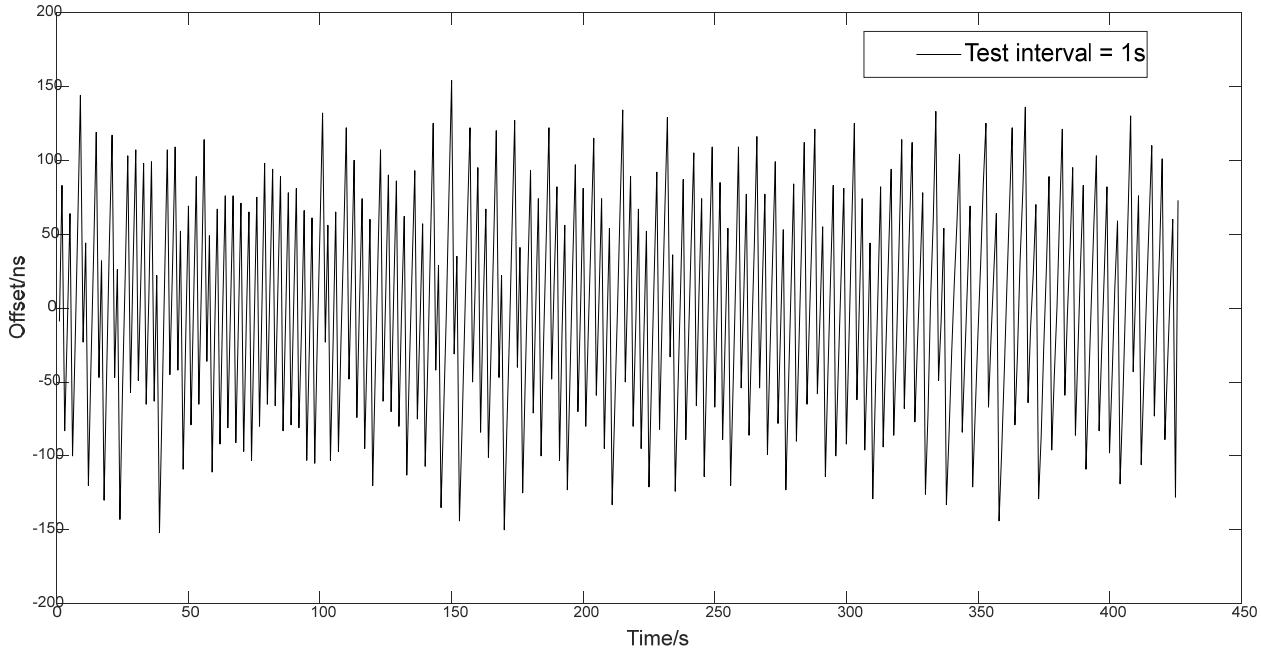


**Figure 5.** Master-slave clock offset

The test result shows that the master-slave clock can carry out normal message communication. The output results are printed on the computer through the serial line, and the master-slave deviation is mostly within 100 nanoseconds.

## 5. CONCLUSION

In a distributed system, each node has high requirements for the progress of the clock, and the synchronization accuracy of some special services reaches the nanosecond level. Based on the IEEE1588 protocol, this paper designs a synchronization system on the basis of an embedded platform. Its synchronization accuracy reaches the nanosecond level, which meets the requirements of most distributed systems.

## ACKNOWLEDGEMENTS

This paper was supported by The Innovation Fund of Postgraduate, Sichuan University of Science & Engineering under Grant Y2019016.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0005

## REFERENCES

[1] H. Guo and P. Crossley, "Design of a Time Synchronization System Based on GPS and IEEE 1588 for Transmission Substations," in IEEE Transactions on Power Delivery, vol. 32, no. 4, pp. 2091-2100, Aug. 2017.

[2] P. A. Crossley, H. Guo and Z. Ma, "Time synchronization for transmission substations using GPS and IEEE 1588," in CSEE Journal of Power and Energy Systems, vol. 2, no. 3, pp. 91-99, Sept. 2016.

[3] F. Girela-López, J. López-Jiménez, M. Jiménez-López, R. Rodríguez, E. Ros and J. Díaz, "IEEE 1588 High Accuracy Default Profile: Applications and Challenges," in IEEE Access, vol. 8, pp. 45211-45220, 2020.

[4] D. Pedretti et al. "Nanoseconds Timing System Based on IEEE 1588 FPGA Implementation," in IEEE Transactions on Nuclear Science, vol. 66, no. 7, pp. 1151-1158, July 2019.

[5] Son, K.J. Chang, T.G. Distributed Nodes-Based Collaborative Sustaining of Precision Clock Synchronization upon Master Clock Failure in IEEE 1588 System. Sensors 2020, 20, 5784.

[6] P. Pandey, B. Pratap and R. S. Pandey, "Implementation of FreeRTOS based Precision Time Protocol (PTP) application as per IEEE1588v2 standards for Xilinx Zynq UltraScale Plus MPSoC devices," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1968-1973.

[7] O. Ronen and M. Lipinski, "Enhanced synchronization accuracy in IEEE1588," 2015 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), Beijing, 2015, pp. 76-81.

[8] P. Loschmidt, R. Exel, A. Nagy and G. Gaderer, "Limits of synchronization accuracy using hardware support in IEEE 1588," 2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Ann Arbor, MI, 2008, pp. 12-16.

[9] J. Han and D. Jeong, "A Practical Implementation of IEEE 1588-2008 Transparent Clock for Distributed Measurement and Control Systems," in IEEE Transactions on Instrumentation and Measurement, vol. 59, no. 2, pp. 433-439, Feb. 2010.