Design of IEEE1588 Based Precision Time Protocol using PSOS

Vipin Kumar Yadav, Arun Kumar, Abhilasha Singh

***Abstract**: The widespread clock synchronization standard, IEEE 1588, is purely master-slave based. The inherent disadvantage is that a failure of the master requires the election of a new one, during which the network nodes cannot be synchronized. The present paper proposes a compatible extension to the standard introducing architecture for fault-tolerant and seamless distribution of time information. In this paper the protocol is implemented using RTOS in a LAN with processor as one node. The processor having real time clock and is programmed for the synchronization between systems. The processor is the grand master clock which is going to generate clock pulses for all other nodes in the LAN. All the other systems in LAN will have a software stack that incorporates the full specification and functionality of the Precision Clock Synchronization Protocol along with a Timing Table and File Table. After the introduction of the architecture the common application of Ethernet is treated to outline the problems arising from this network type. The proposed solution significantly increases the accuracy of the synchronized network in case of failures and additionally provides backup strategies in case of network problems. Network Area Storage (NAS) is implemented as an application between the systems where communication is carried out using PTP.*

***Keywords**: Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (PTP)[1], Proprietary silicon operating system(PSOS), Network Area Storage (NAS)*

I. INTRODUCTION

We use watches and clocks to synchronize ourselves with other persons or procedures. The watch should be accurate – but how accurate depends on the application. Anyone wanting to catch a train should keep an eye on the time to within a minute. At sports events, a hundredth of a second can be decisive and drives in a packing machine need synchronization in the microsecond range. An implicit system time exists when no actual clock is available and the time behavior is determined by processes in the hardware and software. This often suffices in small, closed systems. The system time is explicit when it is represented by a clock. As the industrial systems are complex with computers distributed in space and communicating via networks, the explicit representation of time is often necessary for robust implementations. IEEE-1588-2002, 'Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (PTP)' [1] was designed to serve the clock synchronization needs of industrial systems. The project is based on this protocol.

Clock synchronization is a key factor for many industrial systems. For example, Synchronized clocks are the fundamental requirement for embedded control systems, time-triggered communications, redundancy management, etc. The IEEE 1588 standard or PTP (Precision Time Protocol) provides a precise clock synchronization protocol. for multicast networks. In contrast to NTP (Network Time Protocol), IEEE1588 aims at measurement and control applications and the protocol can provide system-wide synchronization precision in the sub microsecond range. However, the potential drawbacks of PTP are a lack of capability to tolerate faulty master clocks and a selection mechanism that is probably too complicated (and expensive) to be used in low-cost embedded applications.

However, addressing the above mentioned properties of time as a communication variable, fault tolerance [5] has to be included in this kind of real-time networks. To ensure a highly stable reference time, the clock values are distributed from reference nodes to clients. If, due to architectural reasons, all reference nodes are shrunk to one single clock node in the system, this node is a typical single point of failure. Even if a second reference node is provided on a hot standby basis, the system is out of sync during the time-consuming switchover to the new clock reference. The aim of this paper is to enhance master-slave fashioned IEEE 1588 networks in a way that not a designated master is used as reference time node, but to include all nodes with high precision. This allows to get rid of the master which is a single point of failure. The proposed methodology causes the system to run with reduced accuracy in the worst case but keeps the attached nodes synchronized under all circumstances.

II. THE IEEE1588 STANDARD

A. IEEE 1588

The story of simple clock master-slave synchronization with adequate performance in real-time networks is a quite successful one. Since the publication of the IEEE 1588 many standard industrial products adopted the standard. The concept of IEEE 1588 is as simple as effective: After power-up a so-called master election is initiated to elect the most appropriate network node to provide reference time. Clock synchronization is then done subsequently in two steps. First, the delay between master and slave is measured via so-called delay-request and delay-response messages. This

Manuscript received: July, 2012.

Vipin Kumar Yadav, Department of ECE, SRM University, Modinagar, India,
Arun Kumar, Department of ECE , SRM University, Modinagar, India,
Abhilasha Singh, Department of IT, Amity University, Noida, India,

step is needed to enable the slaves to eliminate any transport delay between the two protocol stacks. With this delay information the master sends a sync packet, succeeded by a follow-up packet. This follow-up packet delivers the exact send-time of the previous packet, allowing the slave to cancel out the now known transport delay.

B. Basic IEEE1588 Standard

This is where the Precision Time Protocol (PTP) described in IEEE 1588 comes in. It was developed with the following goals:

• Accuracy to at least microsecond and preferably nanosecond levels.

• Minimal network, computing and hardware resource requirements so that it can be applied to low-end as well as high-end devices.

• Applicable with minimal or no administration to systems defined by a single subnet or at most a few adjacent subnets of a networked system Applicable to common and inexpensive networks including but not limited to Ethernet.

• Applicable to heterogeneous systems where clocks of different capabilities can synchronize to each other in a well ordered manner

• Specified by a standard to promote interoperability and adoption by manufacturers.

III. SYNCHRONIZATION METHODS COMPARED WITH PTP

There have been various previous methods of synchronizing clocks distributed over a network: the most common are the Network Time Protocol (NTP) and the simpler Simple Network Time Protocol (SNTP) derived from it. These methods are quite common in LANs or in the Internet and allow accuracy right down to the millisecond range. Another possibility is the use of radio signals of the GPS satellites. However, this requires relatively expensive GPS receivers for every clock and the appropriate antennae on the roof and the necessary cabling. Although this provides a high precision clock, it is often impractical for reasons of cost and effort.

Table1: Comparison of Synchronization Methods

	1588	NTP	GPS
Spatial extent	A few subnets	Wide area	Wide area
Communications	Network	Internet	Satellite
Target accuracy	Sub-microsecond	Few milliseconds	Sub-microsecond
Style	Master/slave	Peer ensemble	Client/server
Resources	Small network message and computation footprint	Moderate network and computation footprint	Moderate computation footprint
Latency correction	Yes	Yes	Yes
Protocol specifies security	No	Yes	No
Administration	Self organizing	Configured	N/A
Hardware	For highest accuracy	No	RF receiver and processor
Update interval	~2 seconds	Varies, nominally seconds	~1 second

IV. PTP OPERATION

A. Best Master Clock Selection

The most precise clock in the network synchronizes all other clocks. There are two kinds of roles: master (the one, which synchronizes the others) and slaves (those being synchronized). In principle, any clock can play either the master or the slave role. The precision of a clock is categorized by the protocol in classes (stratum). The highest class is an atomic clock that has the stratum value 1. The selection of the best clock in the network is performed automatically using the "best master clock algorithm" [6].

B. Clock Synchronization

PTP's operating principle is to exchange messages consecutively to determine the offset between master and slave but also the message transit delay through the network

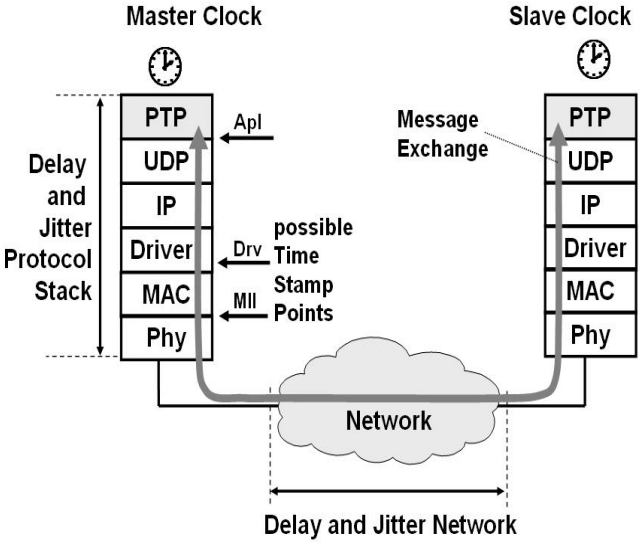


Fig1: PTP message exchange

For the purpose of offset correction, the master cyclically transmits Sync messages to the slave clock at defined intervals (by default every 2 seconds). A time stamping mechanism determines the exact transmission time t1 as precise as possible and sends it down to the slave on behalf of a second message, the Follow-up message (see figure 2).

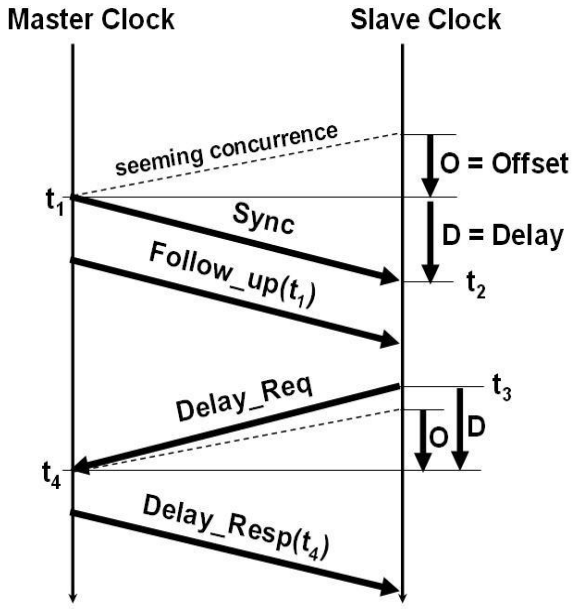


Fig 2: Offset and Delay Measurement

The slave clock measures the exact reception time t2 of the Sync message. On reception of the Sync and the corresponding Follow-up message, the slave clock calculates the offset correction in relation to the master clock. The deviation between master and slave is the transit delay the Sync message has experienced when traveling through the network. Furthermore, consecutive Sync measurements allow the slave's frequency drift to be compensated. Another message exchange is required to measure the delay between slave and master. For this purpose, the slave clock sends a so-called Delay_Req packet to the master. The exact transmission and reception time t3 and t4 of this message are measured.

The Delay_Resp message carries the measured value t4 to the slave, where now delay and offset are calculated out of the four time stamps t1, t2, t3, t4:

Delay + Offset = t2-t1
Delay - Offset = t4-t3
Delay = ((t2-t1) + (t4-t3)) / 2
Offset = ((t2-t1) - (t4-t3)) / 2

Note that the transit delay is assumed to be the same for both directions! The delay measurement is performed irregularly and at larger time intervals (random value between 4 and 60 seconds by default). In this way, the master is not too heavily loaded if it has to synchronize a large number of slaves.

C. PTP Communication

PTP is based on IP multicast communication and is not restricted to Ethernet, but can be used on any network technology that supports multicasting. The scope of this paper is restricted to Ethernet. PTP scales for a large number of PTP nodes because a master can serve many slaves with a single pair of Sync and Follow-up messages. Multicast communication offers also the advantage of simplicity. IP address administration does not need to be implemented on the PTP nodes. For this reason, Delay_Req and Delay_Resp as well as the management messages, which are in fact point-to-point messages, use also multicast addressing. All other nodes but the wanted destination has to filter out these messages.PTP is an application layer protocol, on top of UDP/IP/Ethernet (see figure 3). The use of port number 319, the so-called event port, is reserved for messages that have to be time stamped (i.e. Sync and Delay_Req). This facilitates the message detection logic. All other messages use port number 320.

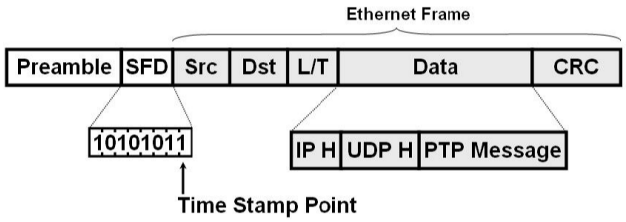


Fig 3: Ethernet encapsulated PTP messages

D. Proprietary Silicon Operating System (PSOS)

pSOS[7] or proprietary silicon operating system is a high performance, modular Real time operating system that is used in a variety of tasks ranging from vending machines to hi-fi satellites. This real time operating system is used in critical tasks where the output time is very critical and also the size has to be very small and performance of the operating system has to be in such a way that the turn around time has to be very less. pSOSystem real-time operating system provides an extensive set of networking facilities for addressing a wide range of interoperability and distributed computing requirements. These facilities include

- **TCP/IP Support** - pSOSystem's TCP/IP networking capabilities are constructed around the pNA+ software component. pNA+ includes TCP, UDP, IP, accessed through the industry standard socket programming interface. pNA+ offers services to application developers as well as to other pSOSystem networking options such as RPC, NFS, FTP, and so forth.
- UDP provides a datagram mode of packet-switched communication. It allows users to send messages with a minimum of protocol overhead. However, ordered, reliable delivery of data is not guaranteed. The IP layer is used for transmitting blocks of data called datagram. This layer provides packet routing, fragmentation and reassembly of long datagram's through a network or internet. Multicast IP support is implemented in the IP layer. The Network Interface (NI) layer isolates the IP layer from the physical characteristics of the underlying network medium. It is hardware dependent and is responsible for transporting packets within a single network. Because it is hardware dependent, the network interface is not part of pNA+ proper. Rather, it is provided by the user, or by ISI as a separate piece of software.

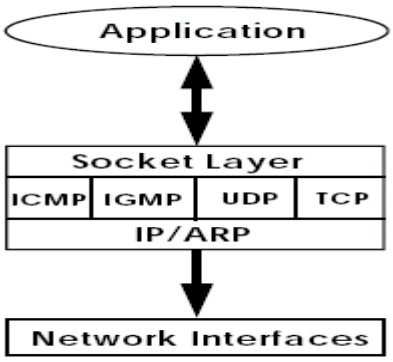


Fig4. pNA+ Architecture

As the total system is implemented using PSOS under which all the processes are divided as tasks. The RTS system has one of the tasks called as Fault Detection Task which is going to execute based on the Time table. The Time Table will maintain all other nodes timing .If due to any problem master fails to achieve good timing then by the fault detection task total system is going to shift into the next best clock in the table. File table is used for Network Area Storage (NAS) application.

V. HARDWARE AND NAS

A. Rabbit Processor

The Rabbit processor is fixed on a module called the RCM2200. The Rabbit core RCM2200 microprocessor module is designed to be the heart of the embedded web server. The RCM2200 features an integrated Ethernet port and provides for LAN and Internet enabled systems to be built easily. The Rabbit core RCM2200 has a Rabbit 2000 microprocessor operating at 22.1 MHz, static Ram, flash memory, two clocks (main oscillator and time keeping), and the circuitry necessary for reset and management of battery backup of the Rabbit 2000's internal real-time clock and the static Ram. Two 26-pins headers bring out the Rabbit 2000 I/O bus lines; address lines, data lines, parallel ports and serial ports.

B. Network Area Storage (NAS)

NAS [4] is a file-level storage technology built on top of SAN or DAS technology. It's basically another name for "file server." NAS devices are usually just regular servers with stripped down operating systems that are dedicated to file serving. NAS devices typically use SMB (server message block) for Microsoft compatibility or NFS (network file system) for UNIX compatibility.

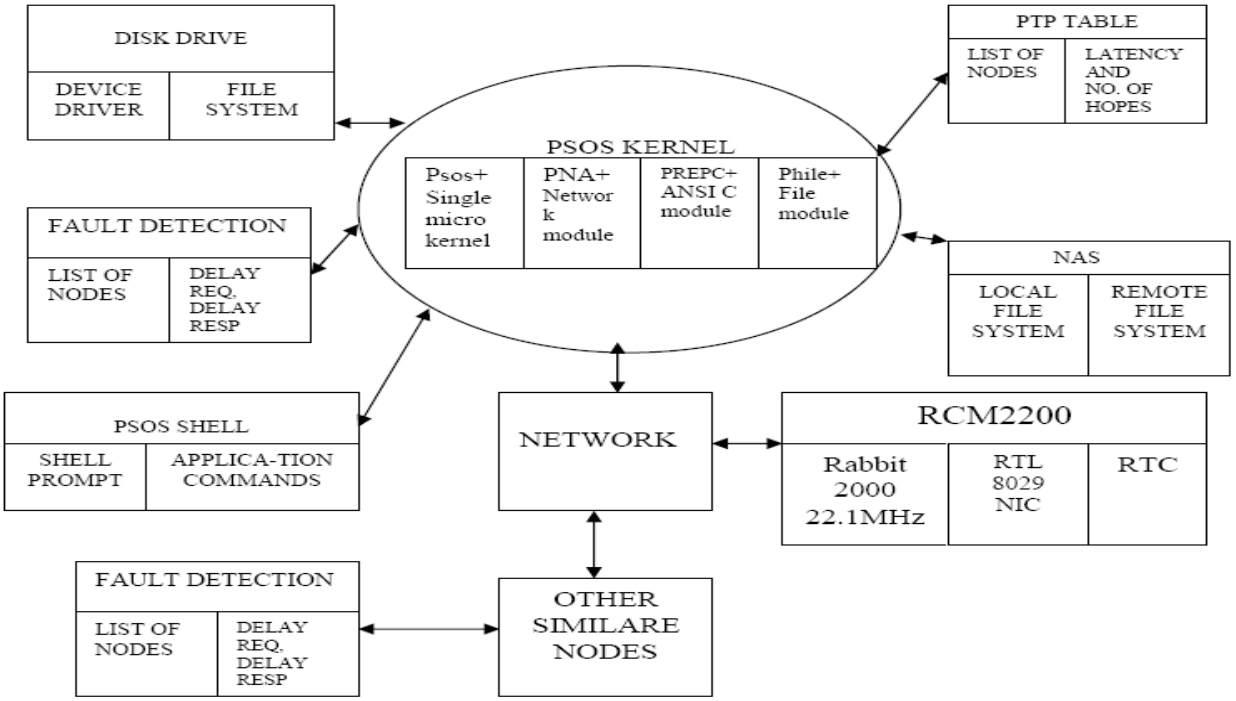
The benefit of a NAS over a SAN or DAS is that multiple clients can share a single volume, whereas SAN or DAS volumes can be mounted by only a single client at a time. The downside to a NAS is that not all applications will support it because they're expecting a block-level storage device, and most clustering solutions are designed to run on a SAN.

VI. IMPLEMENTATION

A. PTP Time Stamping Method

Synchronization accuracy directly depends on time stamp accuracy [6]. As depicted in figure 5 there are different options to take time stamps:

- The most accurate method is to detect PTP frames with hardware assistance. Ingress and egress frames pass the Media Independent Interface (MII), where frames can easily be captured and decoded. Below the MII, data is 4B5B coded, scrambled, and therefore not directly interpretable. The accuracy of this method is limited by the Phy chip timing characteristics.
- Without any hardware assistance, the next best place for time stamping is the network driver. Egress frames are stamped at the very latest moment before the frame is handed over to the MAC controller. Ingress frames are stamped at the entry point of the network interface interrupt service routine. The accuracy of this method is limited by the operating platform timing characteristics (e.g.
- Interrupt latency, CPU performance) and load dependent.
- Stamping on the application layer is best located at the socket interface. It does not require any modification to system software. The influence of protocol stack and load allows only moderate accuracy.



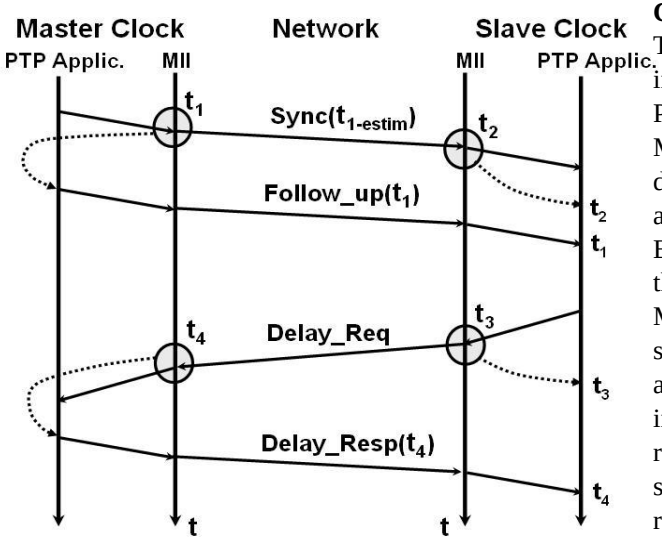


Fig 6: Time stamp transfer (circles indicate time stamping)

B. Implementation hardware

A key aspect for the implementation is a clear interface definition. The system comprises both hardware and software elements. In the hardware element the high precision time is generated while reading the time stamp on the synchronization packets in both transmit and receive directions. The message detector analyses incoming and outgoing packets and detects synchronization packets based on specific values in the packet. For these packets the exact time and the identification of the packet are saved. This can be realized initially using a general purpose Microprocessors or Micro controllers and also using FPGAs, but will in future be integrated into the network components.

C. Software organization

The RTS IEEE 1588 Master and Slave software stack incorporates the full specification and functionality of the Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. Through PTP, multiple devices are automatically synchronized with the most accurate clock found in a packet-based network - typically Ethernet. The RTS protocol stack automatically determines the most accurate clock, otherwise known as the Grand Master Clock. During operation and after initial synchronization, the PTP real-time clocks are constantly adjusted by exchanging timing messages. The RTS implementation uses statistical techniques to further reduce residual fluctuations. Because the RTS IEEE 1588 protocol stack supports the PTP hot-pluggable functionality requirement, devices may join or leave the network at any time. The protocol stack is implemented independently of the operating system and can thus be ported to any of them. The access to the hardware driver is provided in the adaptation layer. Tasks, timers, semaphores and socket interfaces must be adapted for each operating system. These functions are however only needed outside the protocol stack implementation. The protocol makes possible synchronization to within 100ns with hardware support. A pure software solution is, however, also conceivable for IEEE1588. Here, the internal system clock in the computer is used for the time of day delivers a time stamp via the network driver. A protocol implementation purely in software was investigated using Linux. The time stamp was acquired directly at the socket interface. The exercise demonstrated a precision of better than 100µs may be achieved with this implementation alone given appropriate process prioritization.

By further optimizing the drivers it can be assumed that a precision of 10µs and better is achievable.

D. Reconfigurable Hardware/Software-based RTOS

Differently from the normal approach where the design of such RTOS is done offline, the proposed approach suggests the use of new reconfigurable architectures in order to support the development of a hardware/software reconfigurable operating system. In this proposed architecture, the Real-Time Operating System (RTOS) is capable to adapt itself to current application requirements, tailoring the RTOS components for this purpose. Therefore, the system continuously analyzes the requirements and reconfigures the RTOS components at the hybrid architecture optimizing the use of system resources. Hence, the system is capable to decide on-the-fly which RTOS components are needed and also to which execution environment they should be assigned. The target RTOS provide services to the application as a set of hardware and software components. These components can, during run-time, be reallocated (reconfigured) over the hybrid architecture. This system concept is based on the microkernel approach.

E. Performance Chart

Table2: Performance chart

No of Nodes	Number of Packets to Synchronize all Nodes	Time Taken ms
2	4	0.56
3	8	0.72
4	12	0.89
5	16	1.3
10*	36	-
50*	196	-
100*	396	-

F. Performance Criteria

Most IEEE 1588 implementations will have sub-microsecond skew, but actual performance is highly application-specific. For example, the IEEE 1588 protocol does not specify the clock frequency in the master and slaves; thus, lower-frequency clocks have poorer time resolution resulting in less-accurate timestamps in the PTP synchronization messages. Clock stability is another implementation dependency. Clocks based on temperature-controlled crystal oscillators (TCXOs) and oven-controlled crystal oscillators (OCXOs) have higher stability, usually in the parts per billion, compared with clocks using uncontrolled crystal oscillators, with stability in the parts per million. Clocks with lower stabilities will drift apart faster, resulting in more frequent frequency and phase corrections or resulting in larger skews. Another factor is network topology. The simplest network topology -- two devices on a single cable -- provides less network jitter than many devices linked using routers and switches. If more than one subnet is required to increase distance or number of devices, then a network switch with an accurate IEEE 1588 clock, called a boundary clock, becomes the master clock and synchronizes the devices on the subnets. Also, wide variations in network traffic may negatively impact clock skew as the delay correction lags current traffic conditions. Because many factors can degrade skew performance, benchmarking and monitoring actual skew performance over time is recommended.

VII. APPLICATION

The ultimate usage in test and measurement, power distribution, military, and most of all telecommunications awaits the efforts of several standards groups and proven application experience to encourage significant product offerings. A Virtual Storage server for WAN Networks is implemented using PTP for achieving the required synchronization in large networks. The use of a hardware master clock ensures that there is adaptive fault tolerance on the system. Corporate WAN require large amount of data transfer with little delay, the PTP protocol ensures the synchronized transfer of data within the WAN network with sub-microsecond accuracy.

VIII. CONCLUSION

The Precision Time Protocol standardized in IEEE1588 reaches synchronization accuracy within the sub-microsecond range and has further potential for higher precision. It is suited for applications which need a time synchronization of distributed clocks of highest accuracy in a limited network domain in real time applications.

In the future this innovative approach will be extended further to embedded systems which include micro processors, microcontrollers and also FPGAs using concepts like CAN, LIN for their real time applications.

REFERENCES

1. The Case for the Precision Timed (PRET) Machine Stephen A. Edwards_Edward A. Lee November 17, 2006
2. IEEE Std 1588™-2002: "IEEE Standard for a precision Clock Synchronization Protocol for Networked Measurement and Control Systems" http://ieee1588.nist.gov/
3. Systems J. Jasperneite, K. Shehab, and K. Weber. Enhancements to the time synchronization standard ieee-1588 for a system of cascaded bridges. In Proceedings of the 2004 IEEE International Workshop on Factory Communication Systems, pages 239–244
4. An introduction to enterprise data storage, by George Ou|Apr16,2007 http://articles.techrepublic.com.com/5100-10880-6176456.html
5. A Solution for Fault-Tolerant IEEE 1588,Jeff Allan and Dongik Lee, Dependable Real Time Systems Limited The Innovation Centre, 217 Portobello Road, Sheffield, S1 4DP, United Kingdom Phone/Fax: +44-(0)114 223 2301, E-mail: jeff.allan@drts.co.uk
6. IEEE 1588, Standard for a Precision Clock Synchronization Protocol, Prof. Hans Weibel, Zurich University of Applied Sciences, hans.weibel@zhwin.ch, 2005
7. pSOS Product Family, Copyright ã 1997 Integrated Systems, Inc. All rights reserved. Printed in U.S.A.
8. Document Title: pSOSystem System Calls Part Number: 000-5070-004 Revision Date: August 1997
9. IEEE 1588 Real-Time Networks with Hybrid Master Group Enhancements**,** Georg Gaderer, Institute of Computer Technology Vienna University of Technology, Gusshausstrasse 27-29, A-1040 Vienna, Austria, Georg.Gaderer@TUWien.ac.at, Patrick Loschmidt, Thilo Sauter , Research Unit for Integrated Sensor Systems Austrian Academy of Sciences, Viktor Kaplan Strasse 2 A-2700 Wiener Neustadt, Austria
10. {Patrick.Loschmidt, Thilo.Sauter}@OEAW.ac.at

AUTHORS PROFILE

Vipin Kumar Yadav received the M.Tech Degree in EMBEDDED SYSTEM&TECHNOLOGY from S.R.M. Institute of science & Technology, SRM University Chennai in 2009. He is working as Assistant professor in Department of ECE, SRM University NCR Campus modinagar since July 2010. He is currently member of IEEE, ISTE and ISC.

His Research area is DSP, Digital Image processing,

Arun Kumar received the M.Tech Degree in VLSI DESIGN from SRM University Chennai in 2010. He is working as Assistant professor in Department of ECE, SRM University NCR Campus modinagar since July 2010.

His Research area is Low power VLSI, Modeling of semiconductor devices, DSP.

Abhilasha Singh received M.Tech degree in Information Technology from Banasthali University, Rajasthan in 2010. She is working as Assistant Professor in Department of IT, Amity School of Engineering and Technology, Amity University at Noida since July 2010.

Her research area is Communication and Information Security.