Embedded System Design for Network Time Synchronization

So-Young Hwang[1], Dong-Hui Yu[2], and Ki-Joune Li[1]

[1] Department of Computer Science and Engineering
Pusan National University, Pusan 609-735, South Korea
{youngox,lik}@pnu.edu
[2] School of Computer Information Engineering
Catholic University of Pusan, Pusan 609-757, South Korea
dhyu@cup.ac.kr

Abstract. Every computer needs a timer mechanism to keep track of current time and also for various accounting purposes such as calculating the time spent by a process in CPU utilization, disk I/O, and so on, so that the corresponding user can be charged properly. In a distributed system, an application may have processes that concurrently run on multiple nodes of the system. For correct results, several such distributed applications require that the clocks of the nodes are synchronized with each other. Nowadays, time synchronization has been a compulsory thing as distributed processing and network operations are generalized. A network time server obtains, keeps accurate and precise time by synchronizing its local clock to a standard reference time source and distributes time information through a standard time synchronization protocol. This paper describes design issues and implementation of an embedded system for network time synchronization especially based on a clock model. Our system uses GPS (Global Positioning System) as a standard reference time source and offers UTC (Coordinated Universal Time) through the NTP (Network Time Protocol). Implementation results and performance evaluation are also presented.

1 Introduction

Time synchronization is a critical piece of infrastructure for any distributed system. A goal of synchronization procedure is to improve the stability and accuracy of the local clock. Research areas to achieve this goal are generally classified into three parts – standard reference time source, stabilizing of the local clock, and distribution of time information. A network time server acquires and keeps accurate and precise time by synchronizing its local clock to a standard reference time source. It also distributes time information through a standard time synchronization protocol. This paper presents design issues and implementation of an embedded system for network time synchronization. This system uses GPS (Global Positioning System) as a standard reference time source and offers UTC (Coordinated Universal Time) through the NTP (Network Time Protocol).

The remainder of this paper is organized as follows. In Section 2 needs and methods for time synchronization are discussed as related work. Section 3 describes design issues of an embedded system for time synchronization especially based on a clock model. Prototype implementation is also discussed. Next, experiments and performance evaluation are included in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

2.1 General Computer Clock Model

A computer clock usually consists of three components – a quartz crystal that oscillates at a well-defined frequency, a counter register, and a constant register. The constant register is used to store a constant value that is decided based on the frequency of oscillation of the quartz crystal. The counter register is used to keep track of the oscillations of the quartz crystal. That is, the value in the counter register is decremented by 1 for each oscillation of the quartz crystal. When the value of the counter register becomes zero, an interrupt is generated and its value is re-initialized to the value in the constant register. Each interrupt is called a clock tick [9]. Figure 1 shows a general computer clock model.

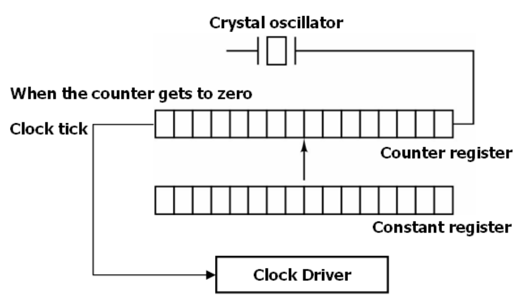


Fig. 1. General computer clock model

A clock always runs at a constant rate because its quartz crystal oscillates at a well-defined frequency. However, due to differences in the crystals, the rates at which two clocks run are normally different from each other. The difference in the oscillation period between two clocks might be extremely small, but the difference accumulated over many oscillations leads to an observable difference in the times of the two clocks, no matter how accurately they were initialized to the same value. Also, a computer clock has limits in accuracy and precision affected by its inherent instability, environment elements, modifications of users, and errors of the system. So a computer clock needs to be synchronized with a standard clock if the computer system requires the precise time processing [7, 9].

2.2 Needs for Time Synchronization

Every computer needs a timer mechanism (called a computer clock) to keep track of current time and also for various accounting purposes such as calculating the time spent by a process in CPU utilization, disk I/O, and so on, so that the corresponding user can be charged properly. In a distributed system, an application may have processes that concurrently run on multiple nodes of the system. For correct results, several such distributed applications require that the clocks of the nodes are synchronized with each other [9]. Nowadays, time synchronization has been a compulsory thing as distributed processing and network operations are generalized. Network operations require time synchronized information to ensure optimal network performance. Often it is not until there is a problem that the lack of time synchronization becomes a key factor in either a failure or the ability to troubleshoot one. In other instances, network processes will not function without time synchronization. Key areas where time synchronization directly effects network operations are [11]:

- Log file accuracy, auditing and monitoring
- Network fault diagnosis and recovery
- File time stamps
- Directory services
- Access security and authentication
- Distributed computing
- Scheduled operations
- Real-world time values

2.3 Approaches for Time Synchronization

There are two approaches for time synchronization: one is that a clock with higher cost, accuracy, and precision replaces a local clock. The other is that the target computer requests services from the external clocks that keep standard time. The former is found in mobile communication systems, for example the PTS (Precise Time Synchronizer) among the IS-95 CDMA (Code Division Multiple Access) base stations. The latter is based on a server-client mechanism that clients request time information from the timekeepers. Although those have variable features according to the internal algorithm and the synchronizing structure, local clocks are generally synchronized to the standard time by obtaining time-related information periodically from the external clock through the predefined communication mechanism [3, 5, 7]. Over the years, the timekeeping community has used many different techniques and system to help them with the task of synchronizing clocks or time transfer. (1) terrestrial communications systems, such as television and telephones (MODEMS); (2) direct radio broadcasts (WWV and WWVH); (3) navigation systems, such as Loran-C and Global Positioning System (GPS); (4) satellite communications system such as two-way satellite time transfer (TWSTT). Most of all, GPS is a versatile and global tool which can be used to both distribute time to an arbitrary number of users and synchronize clocks over large distances with a high degree of precision and accuracy [6]. Since 1985 the Internet has had a well-known, widespread protocol for clock synchronization called NTP, the Network Time Protocol. It is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver or modem. The current version, NTPv3, has been in use since 1992. (NTPv4 is a significant revision of the NTP standard, and is the current development version, but has not been formalized in an RFC.) NTP is able to synchronize clocks with sub-second accuracy across the entire Internet, managing errors from network delays and jitter. NTP has a hierarchical design for clock synchronization. At the top of the tree are the stratum 1 clocks, computers with some source of true time. Other computers synchronize themselves over the network to the stratum 1 clocks, becoming stratum 2 clocks. The process repeats up to stratum 16, which is effectively infinity for NTP [1, 8].

3 Embedded System for Network Time Synchronization

Design issues of an embedded system for time synchronization are as follows: (1) how to acquire the standard time, (2) how to maintain accurate and precise local clock and (3) how to distribute time information.

3.1 Functional Architecture

A network time server consists of a reference clock processing part which acquires the standard time, a timekeeper which maintains the standard time and a packet processor which processes requirements of other clients. A functional architecture of a network time server is represented in Figure 2.

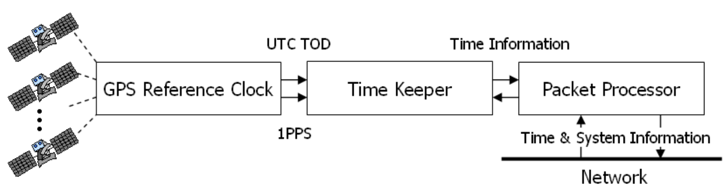


Fig. 2. Functional architecture

GPS is adapted as a reference clock source considering performance and economical efficiency. Global Positioning System (GPS) is not only a navigation system, it is also a time-transfer system. As a time-transfer system it provides stability very close to one part in ten to the fourteenth over one day (1ns/day) [6]. GPS consists of 27 satellites maintained by the US Department of Defense (DoD), each transmitting coordinated "GPS Time" according to its onboard atomic clock. GPS Time differs from UTC only in the absence of the leap seconds which are periodically inserted in UTC. Most GPS receivers automatically take the shift into account using data downloaded from the satellites, so the time reported is UTC. The satellites' onboard clocks are regularly conditioned to match GPS time according to the ground-based reference clock system (actually a large number of high precision atomic time standards). The satellites also broadcast their ephemerides, so their position in space can be accurately calculated as a function of time. The ephemerides also are regularly recalculated and updated. With each satellite's position in space known to high accuracy from its ephemeris, users' receivers can fit for their position and time (x, y, z, t) if four or more satellites are simultaneously in view. Since the GPS satellites are constantly referenced to a national standards laboratory time base, the GPS system provides a simple and inexpensive way to obtain high precision absolute time, synchronized to UTC, without purchasing and constantly re-calibrating a set of atomic clocks [4]. The reference clock processing part needs a GPS engine which processes GPS satellites' signal and a management task which controls and monitors the GPS engine. The mechanism of the timekeeper is specified following subsection in detail. The packet processor handles requirements of clients which need time synchronization. In this paper Internet environment is considered.

3.2 A Clock Model

A clock model for time synchronization should keep stable time and should provide accurate standard time with precise resolution. Figure 3 shows the time keeping mechanism of our system.

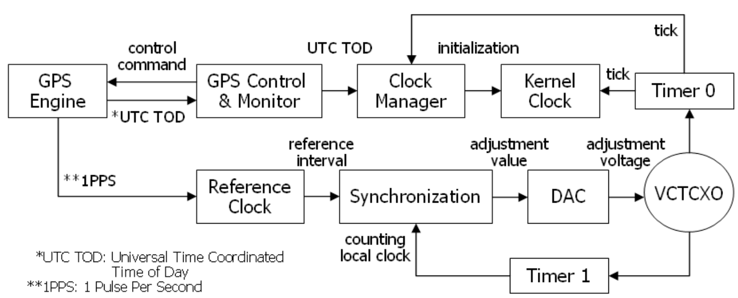


Fig. 3. A clock model

The system clock is initialized by universal standard time received from GPS satellites. It maintains two timers: (1) one is for distribution of time information (NTP) and for the kernel clock which concerns in scheduling, timer functions, and so on, and (2) the other is for synchronizing the local clock to the reference clock source (GPS). The former is affected by the latter directly. Periodic signal of 1 second time interval is adapted as external interrupt to synchronize the local clock. The synchronization module analyzes errors of the local clock and calculates adjustment quantity through the reference clock and the second timer. The adjustment value is adapted to the local clock through DAC (Digital to Analog Converter).

3.3 Prototype Implementation

The selected target platform is a general purpose 32-bit ARM7TDMI microprocessor, developed by Advanced RISC Machines, Ltd. (ARM). The core architecture is based on Reduced Instruction Set Computer (RISC) principles. A ethernet interface is adapted for networking. A GPS engine and a VCTCXO (Voltage Controlled Temperature Compensated Crystal Oscillator) are applied to discipline the local clock. Figure 4 and Figure 5 depict our system architecture.

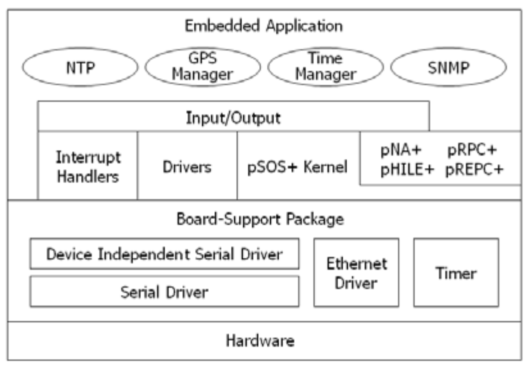


Fig. 4. Software architecture

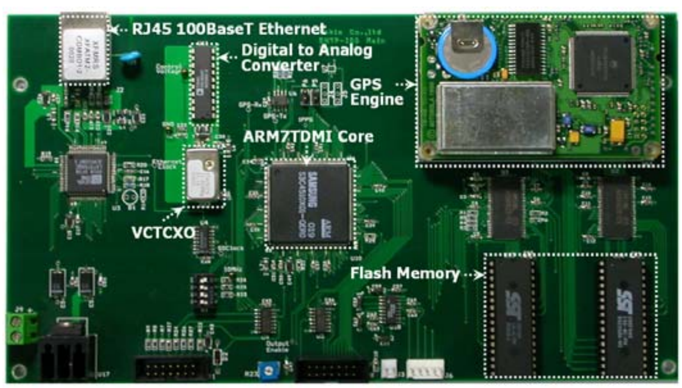


Fig. 5. Hardware architecture

The NTP is applied for standard time synchronization protocol and the SNMP (Simple Network Management Protocol) is adapted for system management. The GPS manager works for GPS engine to control and monitor it. The time manager acquires the standard time and keeps the local clock synchronized to the reference clock. The main system and the GPS engine are connected through the RS-232C serial communication. The GPS engine sends position, status and date message every one second. The proposed clock model is implemented as follows. The system clock and the NTP clock for time distribution are initialized synchronizing to the GPS reference clock. Through the 1PPS reference time interval, system can count its local clock frequency and can analyze errors of the clock [Figure 6].

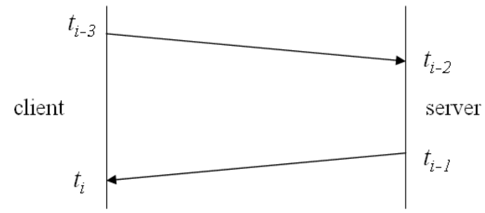
```
begin 1PPS_Interrupt_Service_Routine
    if (tracking_flag == SYSTEM_INIT) then
        call System_Clock_Init;
        call NTP_Clock_Init;
        change flag;
    end if;
    if (tracking_flag == ANALYZE_LOCAL_CLOCK_ERROR ) then
        call Count_Local_OSC_Frequency;
        if (frequency_error > HIGH_ERROR_BOUND ||
            frequency_error < LOW_ERROR_BOUND) then
            call Clock_Adaption_Procedure;
        else continue;
        end if;
    end if;
end
```

Fig. 6. *1PPS* interrupt service routine

4 Experiments

4.1 Plan and Environment

To evaluate our system, ntpq – standard NTP query program is applied. The ntpq utility program is used to query NTP servers which implement the recommended NTP mode 6 control message format about current state and to request changes in that state. The NTP is designed to produce three products: clock offset, round-trip delay and dispersion, all of which are relative to a selected reference clock. Clock offset represents the amount to adjust the local clock to bring it into correspondence with the reference clock. Round-trip delay provides the capability to launch a message to arrive at the reference clock at a specified time. Dispersion represents the maximum error of the local clock relative to the reference clock.



Timestamps exchanged with possibly many other servers are used to determine individual round-trip delays and clock offsets relative to each server as follows. Number the times of sending and receiving NTP messages as shown below and let i be an even integer. Then $t_{i-3}, t_{i-2}, t_{i-1}, t_i$ are the values of the four most recent timestamps as shown. The round-trip delay δ_i and clock offset θ_i of the client relative to server is [10]:

$$\delta_i = (t_i - t_{i-3}) - (t_{i-1} - t_{i-2}) \,,$$

$$\theta_i = \frac{(t_{i-2} - t_{i-3}) + (t_{i-1} - t_i)}{2} .$$

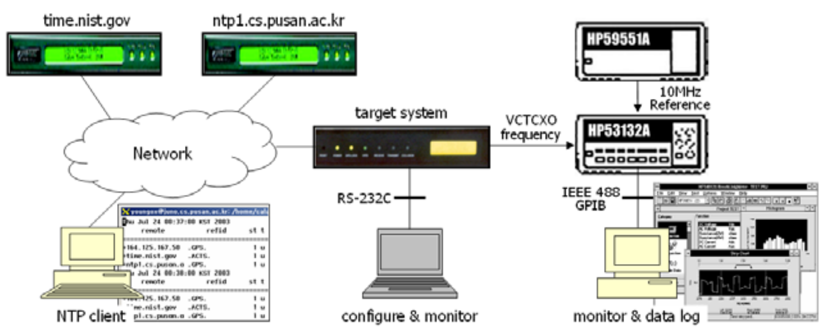


Fig. 7. Experiment environment

A test environment of the network time server is represented in Figure 7. We use the HP53132A universal counter as a electronic frequency counter and the HP59551A GPS synchronizer as a reference source to the HP53132A. A Linux workstation works as a NTP client. We equipped a monitoring PC with the IEEE 488 GPIB interface to record events from the universal counter. The PC can operate as the monitoring purpose and/or data logging purpose. The NTP client was synchronized through the target system and other two public time servers to analyze accuracy of the system. Frequency of the target system was also measured.

4.2 Experiment Results

Frequency measurement result shows that it traces the nominal frequency in about ±0.5Hz error boundary [Figure 8].

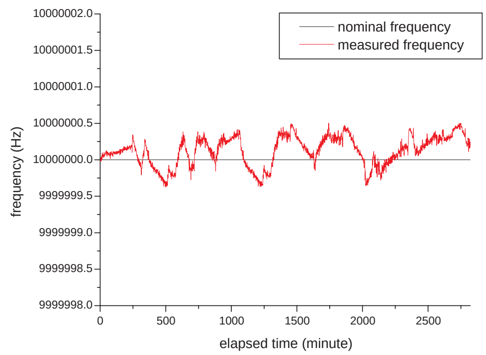


Fig. 8. Variation of frequency

Accuracy is the degree of conformity of a measured or calculated value to its definition. Accuracy is related to the offset from an ideal value. In the world of time and frequency, accuracy is used to refer to the time offset or frequency offset of a device. For example, time offset is the difference between a measured on-time pulse and an ideal on-time pulse that coincides exactly with UTC. Frequency offset is the difference between a measured frequency and an ideal frequency with zero uncertainty. This ideal frequency is called the nominal frequency [12]. The relationship between accuracy and stability is illustrated below. Our target system has accuracy but it is needed for improvement on stability.

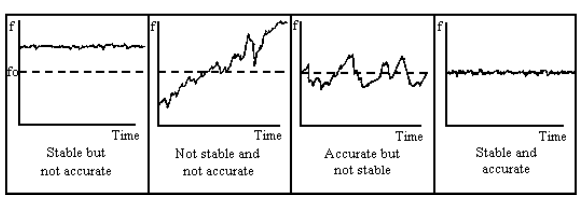


Fig. 9. Relationship between accuracy and stability [12]

The variations of the clock offset and the dispersion represent accurate and stable operations of the target system [Figure 10], [Figure 11].

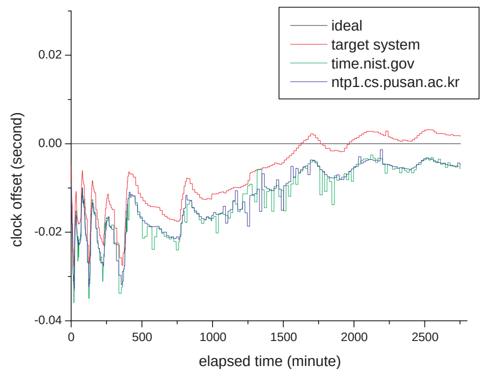


Fig. 10. Variation of the clock offset

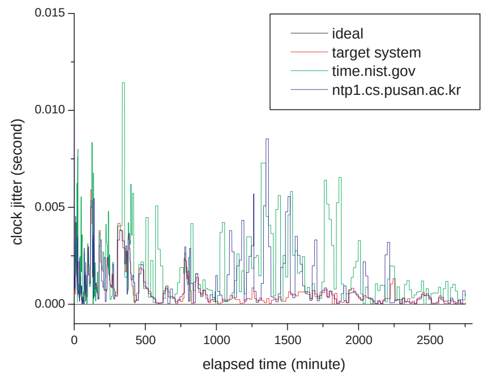


Fig. 11. Variation of the dispersion

5 Concluding Remarks

A computer clock has limits in accuracy and precision affected by its inherent instability, environment elements, modifications of users, and errors of the system. So the computer clock needs to be synchronized with a standard clock if the computer system requires the precise time processing. The purpose of synchronizing clocks is to provide a global time base throughout a distributed system. Once this time base exists, transactions among members of distributed system can be controlled based on time. For example, the management of redundant data in a real time fault-tolerant system can be simplified if the processors are synchronized. Nowadays, time synchronization has been a compulsory thing as distributed processing and network operations are generalized. This paper described design issues and implementation of a network time server which obtains and keeps standard time by synchronizing its local clock to a standard reference time source and distributes time information. The system uses GPS as a standard reference time source and offers UTC through the NTP. As a future work, it is needed that research on a clock model and a clock adaptation mechanism to maintain more accurate and precise time. Moreover, distributed, wireless sensor networks make extensive use of synchronized time, but often have unique requirements in the scope, lifetime, and precision of the synchronization achieved, as well as the time and energy required to achieve it [2]. Existing time synchronization methods need to be extended to meet these new needs.

Acknowledgment. So-Young Hwang would like to thank Professor Youngho Kim, the only mentor of hers, who passed away.
This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce, Industry and Energy of the Korean Government.

References

1. Mills, D.L.: A brief history of NTP time: memoirs of an Internet timekeeper, ACM SIGCOMM Computer Communication Review, Volume 33, Issue 2, pp.9–21, 2003.
2. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-Sync Protocol for Sensor Networks, Proceedings of ACM SenSys pp.138–149, 2003.
3. Levine, J.: Efficient time transfer using the Internet, Proceedings of Frequency Control Symposim and PDA Exhibition, pp.522–529, 2002.
4. Berns, H.C., Wilkes, R.J.: GPS time synchronization system for K2K, IEEE Transactions on Nuclear Science, Volume 47, Issue 2, Part 1, pp.340–343, 2000.
5. Yu, D.H., Hwang, S.Y., Seong, S.Y., Kim, Y.H.: An analysis of error factos in network time server, Proceedings of the GNSS Workshop, pp.159–162, 2000.
6. Lewandowski, W., Azoubib, J., Klepczynski, W.J.: GPS: primary tool for time transfer, Proceedings of the IEEE, Volume 87, Issue 1, pp.163–172, 1999.
7. Jun, S.M., Yu, D.H., Seong, S.Y., Kim, Y.H.: A time synchronization method for NTP, Proceedings of the RTCSA, pp.466–473, 1999.
8. Minar, N.: A survey of the NTP network, MIT Media Lab., 1999.
9. Sinha, P.K.: Distributed Operating Systems: Concepts and Design, IEEE Computer Society, pp.282–292, 1997.
10. Mills, D.L.: Network Time Protocol (Version 3) Specification, Implementation and Analysis, RFC1305, 1992.
11. Skoog, P.: The Importance of Network Time Synchronization, TrueTime Inc.
12. Time and Frequency from A to Z, http://www.boulder.nist.gov/timefreq/