Clock Synchronization in Distributed Area

[1]Vijay Masne, [2]Sagar Badhiye , [3]Nilesh Sambhe , [4]Rashmi Phasate

[1] Computer Technology Department, YCCE
Nagpur, Maharashtra 441110, India

[2] Computer Technology Department, YCCE
Nagpur, Maharashtra 441110, India

[3] Computer Technology Department, YCCE
Nagpur, Maharashtra 441110, India

[4] Electrical Engineering Department, GHRIETW
Nagpur, Maharashtra 440028, India

Abstract - Accurate clock synchronization is difficult to get into distributed environment. Since last decade most of the transactions are online viz. Online banking applications, database queries and real time applications. Time synchronization in a distributed system is important because time based queries can be answered only if all the distributed system has a common notion of time. In distributed system the clocks do not remain well synchronized without periodic synchronization. To maintain the global time the clocks of the nodes must resynchronized periodically. The aim of this paper is to study existing time synchronization approaches and analyze the need of a new class of clock synchronization protocol that is scalable, topology independent, fast convergence, and less application dependent in a distributed environment. The review`s presented in this paper are on the basis of different techniques for clock synchronization proposed by various researchers . This work will help in selecting appropriate methods of clock synchronization in a distributed environment.

***Keywords* - Clock Synchronization, Clock Accuracy, Clock drift, Distributed environment, Group synchronization, Network Delay, Synchronization issues**.

1. Introduction

To synchronize the local clocks of nodes in a distributed environment have been extensively studied in the last decade and yet there is no specific clock synchronization scheme available to achieve a higher order of accuracy with greater scalability of protocols and techniques. In several of the existing measurement schemes, Precision Time Protocol (PTP) [1], Network Time Protocol (NTP) [2], Internet protocol (IP) [4], Delay measurement time synchronization (DMTS) [4], Flooding time synchronization protocol (FTSP) [4], Delay measurement time synchronization protocol (DMTS) [5], Delay Tolerant Networks (DTN) [7], Delay Estimated Clock Sampling Network Synchronization (DE-CSNS) [12] Reference broadcast synchronization (RBS) protocol [13], are used with the objective to perform synchronization. Clock synchronization is a vital need in Delay Tolerant Networks (DTN) for providing accurate timing

information from physical environments [2]. For making a network to be synchronized in terms of time, multiple modes of operation are required some of them are multiple access modes, sleep mode, wake up mode, data integration mode, etc. To improve the performance of scattered environment, critical requirements for synchronization is essential to enhance accuracy in a network [1]. Clock synchronization is a technique by which all the nodes deals with the real time system, a network runs at the same time with any kind of clock drift between the nodes. The main goal of clock synchronization is to manage multiple unrelated processes running on different machines. It should also be able to make consistent decisions about the ordering of events in a system.

It will ensure that all machines report at the same time, regardless of how imprecise their clocks may be or what the network latencies are among the machines. Whereas in scattered networks, since the problem with internal hardware of the computer system may give a wrong clock time, the clocks have errors. The clock may drift due to environment changes, such as temperature, internal aging, etc. In a network, it is well known fact that, all the nodes are distributed in nature and there is no fixed infrastructure or central node to make all the nodes synchronous; thus clock synchronization is a critical issue. From the previous work [2], [7], [13], it is observed that, nodes in a distributed environment may not be synchronized well initially, when the network is deployed. The nodes may be turned on at the different time and their clocks may be running according to different initial values.

The result of events on specific node may also affect the clock, for example, all the broadcast channels have some interval of time between the broadcast start and display to the user. So to reduce a time delay occurring in broadcasting, broadcasters can broadcast and the users can watch the display at nearly the same time.To achieve the real time of the network, several researchers have proposed a variety of techniques [4], [5], [7]. Some of them have really put forth the delay problem in an efficient way. This work explores current research in time synchronization problem due to clock offset by analyzing related work on existing clock synchronization schemes and provides a comprehensive analysis for future researches. The technical concept, performance analysis and their claims are summarized. The paper is organized as follows section II synchronization issues, section III the brief reviews of the existing techniques in clock synchronization are described. In Section IV; Group Synchronization is discussed. In Section V; Conclusion is followed by future work.

2. Clock Synchronization Issues

Clock synchronization problems and various issues in designning the synchronization protocols are defined in [2]. The major issue in a distributed environment is of the connected nodes dealing with their local clocks which may differ. The problem may arise during data transmission. This can be resolved if the application nodes to get synchronized with each other This leads to relatively high accuracy in a network.

3. Existing Techniques

Time synchronization is a key for many applications and operating systems in distributed computing environments. Many protocols have been proposed and used for time synchronization to improve the communication for resource sharing in the distributed environments. Most of the work in clock synchronization is based on traditional Reference History Based Protocol, message passing technique and Delay measurement. What follows next is recent work in this field [1], [2], [4], [8].

3.1 Reference History Based Protocol

This technique is an agreed standardized way of performing a task. The process is repeatable and reproducible. Every time two different computers or programs need to agree on how they will communicate information between them. Every time any file is to be downloaded protocols are required, otherwise the computers will be unable to agree on which file should be downloaded. NTP and PTP require measuring Round-Trip Time (RTT) to perform clock synchronization, and even more restrictive [1], [2], they require both ways (from source to destination and from destination to source) to have equal one way delay to function. It is difficult to expect this from a network such as Internet [8]. To calculate cyclic paths properly each one-way delay, measures round-trip delays between the nodes. PTP is more accurate than NTP but it requires more round trips from source to destination which will raise the problem of delay in clock time. It will also work on a wide network whereas NTP does not deal with a wide area. It is based on the assumption that PTP achieves clock synchronization. PTP obtains the accuracies in the milliseconds or even microsecond range with over hundred nodes. The server continuously sends the synchronization packet. The precise timing information is sent in a two-step process as shown in Fig. 1. First, the packets are time stamped with an estimate of the sending time. The actual sending time is sent with a so-called follow-up packet in the second step [13].

The NTP clients synchronize their clocks to the NTP time servers with accuracy in order of milliseconds by statistical analysis of the round-trip time. The time servers are synchronized by external time sources, typically using GPS. The NTP has been widely deployed and proved to be effective, secure and robust in the internet [2], [5], [14]. Time caused by NTP is more, that can introduce several undreds of milliseconds delay at each hop. Therefore, without further adaptation,

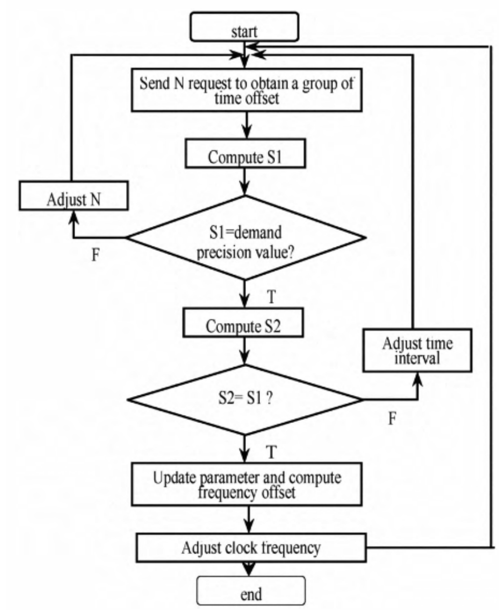

Fig. 1. Flow chart of clock synchronization [15]

NTP is suitable only for WSN applications with low precision demands. There were other schemes such as RBS [13] which eliminate the uncertainty of the sender by removing the sender from the critical path. Many of the time synchronization protocols uses a sender to receiver synchronization method where the sender will transmit the timestamp information and the receiver will synchronize. RBS is different because it uses receiver to receiver synchronization. The idea is that a third party will broadcast a beacon to all the receivers. The beacon does not contain any timing information; instead the receivers will compare their clocks to one another to calculate their relative phase offsets. The timing is based on when the node receives the reference beacon whereas to provide protection against interference; authenticate that data from a trusted source. The protocol had a number of qualities, including: round-trip delay, consistency of the delay, round-trip error, the accuracy of the server's clock. NTP exchanges time with several clients and servers, a process can determine which server to favor. The preferred ones are those with a lower stratum and the lowest total filter dispersion. A higher stratum which may be less accurate time source may be chosen if the communication to the more accurate [2], [12].

For monitoring system along with a better security a virtual machine monitor (VMM) supports a new data transfer mechanism [3]. It is preferable to add extra functionalities to the VMM, which may introduce security vulnerabilities and complicate the implementation of the VMM. Serialization/deserialization is a standard operation in RPC systems. This operation is expensive because it involves a large amount of computation for looking up data tables, walking the data structure to pack them properly. In a typical RPC, the serialization /deserialization operations commonly occur, resulting in enormous computation overhead, too many system calls are involved in each RPC operation.

Traditional RPC systems have two inherent problems. First, their performance is architecturally limited by the cost of invoking system calls, copying data between the user space and the kernel space, and possible thread rescheduling. Second, in VMs some system calls must be trapped and handled by the VMM, leading to significant context-switch overhead. Throughput of RPC was heavily dependent on the type of the data transferred, over the network clients with a clock time for synchronization. To overcome this overhead and long distance synchronization over a long period of time CDR could be subsequently used to adjust the clock and timing of Femtocell Base Station (FBS) and to ensure that the overall network achieved synchronized time.
For faster convergence and more accuracy, synchronization, the algorithm was combined with decision fusion or gossip averaging algorithms [4]. As a result, synchronized time was maintained over a longer period between the nodes and within the network. Internet protocol (IP) is a backbone for a network but it suffers from delays due to the latencies in the protocol.

The timing estimation accuracy of such an approach can be adversely affected by the different downlink and uplink speeds of the backbone network. Unlike GPS signals, TV signals do not have worldwide coverage and the design of TV receivers was zone dependent. Moreover, traditional TV signals available in most areas around the world do not contain accurate time synchronization information. So, to overcome from that drawback they implemented a technique called as a clock drift ratio (CDR) to achieve and maintain synchronization for a longer period of time throughout the network while, it should not dependent on an external source for timing synchronization [2]. The timing drift and offset in sleep clocks may not always obey has given random distributions. Therefore, they model the long-term sleep clock synchronization as a closed-loop control problem. Here the local time was treated as a feedback variable and the clock uncertainty as a disturbance [1].

Another time synchronization class of protocol is FTSP [5]. This protocol is similar to TPSN, but it improves the shortcomings of TPSN. It is similar in the fact, that it has a structure with a root node and all nodes are synchronized to the root. The root node will transmit the time synchronization information with a single radio message to all participating receivers.
The sleep clock source was usually a low-frequency crystal oscillator with poor frequency stability. Furthermore, the working period, namely, the synchronization period, was destined to be very long in the on–off working mode. Hence, sleep clocks may suffer perturbation from the clock model and external disturbances in such a long period. To achieve accurate drift compensation, they estimated uncertainty. Since DMTS and FTSP used the same method to do offset compensation as Feedback based synchronization (FBS) [13], [14], the performance of FBS in compensating clock drift to improve synchronization accuracy.

3.2 Delay Measurement Based Techniques

The technique is based on the Delay measurement of the clock synchronization. It is possible to measure the transmission time of the token. This time can be assumed to be constant. The assumption holds, synchronization to be more accurate, because variations in message transmission delay introduce error in the clock offset calculation. This measured data can help in decision making in traffic routing and fault detection. The resulting delay can then be used as a true measurement for clock synchronization of the network after removing such clock skew.

The technique called Next Generation Network (NGN), are used for measuring delay in the network. NGN transports packet for synchronization. In this research the delay was reduced but this technique was not enough to provide accuracy in a real time network. It aligns the time and frequency scales of all the clocks by using the communications capacity of links between nodes [6]. In the research of the Delay measurement technique the Asynchronous Diffusion (AD) protocol [4] takes a longer time to converge than the Distributed asynchronous Clock Synchronization (DCS) protocol. DCS protocol can achieve faster convergence speed and provides more accurate timing [8], [9]. Their research introduced a protocol that was fully distributed, so that all nodes independently execute exactly the same algorithm without the need of reference nodes. The Timestamp Transformation Protocol (TTP) [4] solves the temporal ordering problem in sparse adhoc networks. The protocol does not synchronize clocks, but transforms message timestamps at each node to its local timestamp with some error bound as a message moves from hop to hop. Simulation results show that the clock inaccuracy increases linearly with time and the number of hops. The DTP achieves a clock estimation error lower than the NTP by explicitly estimating the relative clock frequency using back-to-back messages with a controllable interval in between. However, the DTP is a reference node clock synchronization which assumes at least one time server in the network, and it does not work in a distributed environment where there is no reference node to spread the correct reference clock information. Double-pairwise Time Protocol (DTP) provides time synchronization in DTNs with a modified NTP [7].

The Clock Synchronization Architecture of Network for Internet has good adaptability, to avoid the network transformation. It facilitates management and organization of the clock synchronization system. It ensures the security and accuracy of the clock synchronization [8]. The Scheme to Measure One-way Delay Variation (OWDV) with detection and removal of clock skew is used to travel packets between the source and destination host. This can easily be measured if clock synchronization is performed between these two hosts and this method depends on the clock skew adjustment. It can work on public and private VoIP networks [9]. The Layered Diagnosis and Clock-Rate Correction for the TTEthernet Clock Synchronization Protocol introduced new distributed algorithms which can be implemented as layers on top of the TTEthernetclock synchronization protocol.

The formal framework is based on a model checker for infinite data types. The interactions of the components were hard to trace over the network. The synchronization quality decreases with the number of faulty components and the severity of their failure modes. The TTEthernet clock synchronization algorithm is inherently fault tolerant. The rate correction algorithm records the clock state correction values for a configurable number of integration cycles. Using this technique, faulty devices were detected, in particular network, and if found remove them. By doing so, the failure mode of a faulty node was transformed from an inconsistent omission failure mode to a fail silent failure mode. This could improve the precision of the system which will allow to realistically modelling real-time clocks. In Clock rate correction algorithm a delay is measured in network and gets corrected with the server clock time. An OWDV distributed algorithm was notoriously difficult, particularly in the case of fault tolerant algorithms [10].

Security issues over an internet, enemy can destroy the process of clock synchronization. So there are problems about how to ensure the security of the clock synchronization of the Internet. To resolve the above problems effectively, it needs to establish reasonable clock synchronization system architecture of the Internet [8]. The basic mechanism of clock synchronization over the internet was to access synchronized clock servers by the Internet. The synchronized servers synchronize the clock of connected nodes in their networks. The advantages of the organization level architecture of clock synchronization were: it is easily organized, managed and controlled, and it can improve the security of clock synchronization. It has good adaptability, so it can avoid the network transformation [7], [9].

3.3 Message Passing Based Techniques

Timing information can be exchanged among each node or by the way of designated time servers. Time server simply provides a reference time, which can be queried by other nodes.These clients update their clocks accordingly. Alternatively, servers can pull a set of nodes for their respective time and then determine a new common time, which is again sent to the clients. In this case, the server adjusts its own time, too. Some of the implemented solutions are as follows.

In the Message exchange mechanisms and statistical signal processing techniques, the two-way message exchange is used as a classical mechanism for exchanging timing information between two adjacent nodes. The clock synchronization in a distributed environment can be regarded as the process of removing the effects of random delays from the timing message transmissions sent across connected nodes. Clock synchronization in a network can be achieved by transferring a group of timing messages to the target sensors. The timing messages contain information about the time stamps measured by the transmitting sensors [11]. The main problem with this technique is data loss because of the aging of internal hardware. All the nodes are in distributed state, there is no fixed infrastructure. All the existing methods need to send packets and using the time information to complete the synchronization, as a result of the delay experienced by packets sent, the synchronization accuracy greatly reduced. In virtue of network delay estimated, the program can effectively reduce the synchronization error compared with other methods [12].

The fundamental limits on synchronizing clocks over networks characterized are fundamentally feasible and infeasible in synchronizing clocks over wired or wireless networks. The skew can be determined correctly, because of the determination of unknown clock offsets, and the link delays are in general impossible. A sender can predict exactly the time its packet will be received. The nodal skews can determine correctly but only delay differences between neighbouring communication links with a common sender [13].These are the two-way message exchange (or sender– receiver synchronization), the one-way message dissemination and the receiver–receiver synchronization [9]. By exploiting Maximum likelihood estimator (MLE) analysis and Cramer Rao bound (CRB); the synchronization accuracy at any node in the network can be predicted and used to determine how many data samples are necessary to achieve certain synchronization accuracy.

This might be an important feature for applications that require tight synchronization accuracy.
The fundamental role of signal processing techniques was demonstrated in the context of clock synchronization in WSNs. The clock synchronization protocol provides some directions to the necessary ingredients for devising an optimal estimator operating under an unconventional environment. However, centralized signal processing techniques can only help solving the problem of node-to-node synchronization and possible ad hoc extensions to network-wide synchronization. The next important step was the application of decentralized signal processing techniques (e.g., Distributed estimation and detection) to the clock synchronization problem, and this naturally results in desirable distributed clock synchronization algorithms. Furthermore, distributed signal processing techniques reveals the optimal way of message passing, thus saving unnecessary communication overhead.

The two problems are coupled and it is beneficial to jointly solve the two problems at the same time. Furthermore, the problem of devising joint synchronization and localization algorithms assuming non line-of-sight transmission was also investigated. With these promising results, it can be believed that the signal processing techniques exploited in localization applications could be further applied in the context of clock synchronization and vice versa.

All the existing methods need to send packets and using the time information to complete the synchronization. As a result of the delay experienced by packets sent, the synchronization accuracy greatly reduced. So although Clock sampling mutual network time synchronization CSMNS method achieved a certain precision in theory, it does not consider the various types of delay experienced by packets. Hence results in differences of the received time information. This is not suitable for application in real networks. To solve problems in the existing methods that depended on special nodes and synchronization accuracy, not so high due to the network delay, a new method DE-CSNS (Delay Estimated Clock Sampling Network Synchronization) based on CSMNS was proposed. Compared with other methods, DE-CSNS does not depend on particular nodes [8], [11], the low frame cost was suitable for multi-hop networks, and the delay measurement method was effective in reducing the synchronization error.
The use of mutual synchronization method avoided these problems. But because of the existing delay, mutual synchronization accuracy was not so high. On the basis of mutual synchronization method, the program measures network delay in the networking process, to avoid selecting the master node and synchronization, accuracy was better than other mutual methods. A synchronization method that initiating nodes measure network delay as nodes send access request messages in the access request process, and save the delay value. After the network gets conducted every node uses mutual synchronization and every node can adjust its clock spontaneously only with the synchronization information from neighbouring nodes [5].

4. Group Synchronization

All current approaches employ the technique of pairwise synchronization but this is insufficient in cases where a group of nodes needs to be synchronized for instance in transaction application. In a group synchronization, all receivers respond with their clock offset to the server. The server has one measurement to relate each pair of nodes which is either form a sender-receiver pair or a receiver-sender pair [8]. Another problem in Synchronization algorithm duplicates detection of events. The time of an event helps to determine, time of event will occur from both ends. All these applications will function accurately only if the synchronization error between nodes in a group is bounded [11], [14]. All the existing techniques concentrates on establishing pairwise relationships between a pair of nodes at a given instant of time. However, there is need of developing a unified framework that uses one optical server to synchronize all nodes in the group or wide distributed environment at the same instant of time.

5. Conclusion

In this paper, the existing techniques of clock synchronization have been studied. Most of the recent techniques are based on Reference History Based Protocol, Delay measurement and Message Passing. The shortcomings of most techniques are its insufficient exposure to real time clock synchronization. In future a system can be developed to provide platform independency for clock synchronization. So that, in distributed network, the program can effectively reduce clock synchronization error and delay in the network. It will very helpful in real time application.

References

[1] Georg Gaderer, Patrick Loschmidt, and ThiloSauter, "Improving Fault Tolerance in High-Precision Clock

Synchronization", IEEE Trans. on industrial informatics, vol.6, no.2, pp. 206-215, May 2010.

[2] Paul Krzyzanowski, Clock Synchronization, Rutgers University–CS 417: Distributed Systems 2000-2009, pp.1-14.

[3] Hao Chen, Lin Shi, Jianhua Sun, Kenli Li and Ligang He, "A Fast (Remote Procedure Call) RPC system for Virtual Machines", IEEE Transactions on parallel and distributed systems, pp. 1-11, 2012.

[4] Hani Mehrpouyan, Steven D. Blasting, Tommy Svensson, "New Distributed Approach for Achieving Clock Synchronization in Heterogeneous Networks", IEEE Globecom, 2011 proceedings.

[5] Jiming Chen Qing Yu, Yan Zhang, Hsiao Hwa Chen and Youxian Sun, "Feedback-Based Clock Synchronization in Wireless Sensor Networks: A Control Theoretic Approach" IEEE Transactions on vehicular technology, vol. 59, no. 6, pp. 2963-2973, July 2010.

[6] Stefano Bregni, Ravi Subrahmanyan, "Synchronization over Ethernet and IP in Next-Generation Networks". IEEE Communication Magazine, pp. 130-131, Feb 2011.

[7] Bong Jun Choi, Halo Liang, Xuemin Shen, Fellow, and Weihua Zhuang, "Distributed Asynchronous Clock Synchronization in Delay Tolerant Networks", IEEE Transactions on parallel and distributed systems, vol. 23, no. 3, pp. 491-504, March 2012.

[8] Junwei Lv, Xiaohu Yuan and Haiyan Li, "A New Clock Synchronization Architecture of Network for Internet of Things", International Conference on Information Science and Technology, pp. 685-688, March 2011.

[9] Makoto Aoki, Eiji Oki, and Roberto RojasCessa,"Scheme to Measure One-way Delay Variation with Detection and Removal of Clock Skew", International Conference on High Performance Switching and Routing, pp. 159-164, 2010.

[10] Wilfried Steiner, Bruno Dutertre, "Layered Diagnosis and Clock-Rate Correction for the TTEthernet Clock Synchronization Protocol", IEEE Pacific Rim International Symposium on Dependable Computing, pp. 244-253, 2011.

[11] Yik Chung Wu, Qasim Chaudhari, and Erchin Serpedin, "Clock synchronization in wireless sensor network", IEEE Signal Processing Magazine, pp. 124-138, January 2011.

[12] Lu Qi, Wu Chen, "A Clock Synchronization Method for Ad hoc Networks", pp. 3614-3617, 2011.

[13] Nikolaos M. Freris, Scott R. Graham and P. R. Kumar, "Fundamental Limits on Synchronizing Clocks Over Networks", IEEE Transactions on Automatic Control, vol. 56, no. 6, pp. 1352-1364, June 2011.

[14] Prakash Ranganathan, and Kendall Nygard "Time synchronization in wireless sensor networks: a survey", International journal of Ubicomp (IJU), vol. 1, no. 2, pp. 92-102, April 2010.

[15] Bin Zhao, Min Zhang, Peng Zhai and Haifeng Wu "Design of network time synchronization algorithm based on Frequency adjustment", International Conference on Advanced Computer Theory and Engineering(ICACTE) China, vol. 1, pp. 59-63, August 2010.