International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 03 – Issue 04 || April 2018 || PP. 14-17

# Clock Synchronisation in Distributed Systems: A Review

M. Leela, D. Manoj Kumar, G. Bhavana
*Department of Computer Science & Technology, KPR Institute of Technology, Hyderabad-500088, India*

**Abstract:** Time synchronization is a critical piece of infrastructure in any distributed systems. Since resource sharing is the vigorous objective of any distributed system, proper synchronization should be mandatory in distributing the resources. Also, synchronization is necessary to have resources available when a request is generated. In addition to frequency, synchronization is based on time. We can have any type of distributed systems i.e., wired or wireless. Synchronization can be achieved through various approaches of dealing. In this paper, we perform a comprehensive study about the most common existing time synchronization approaches and stress for having a new approach of synchronization.

**Keywords:** distributed system; clock synchronization; resource sharing; wired communication

## 1. Introduction

Every computer has a physical clock that counts oscillations of a crystal. This hardware clock is used by the computer's software clock to track or know the current time. However, the hardware clock is subject to change -- the clock's frequency varies and the time becomes inaccurate. As a result, two clocks will be slightly different at any given time. The difference between two clocks is called their skew [5].

There are several methods for synchronizing physical clocks. External synchronization means that all computers in the system are synchronized with an external source of time (e.g., a UTC signal). Internal synchronization means that all computers in the system are synchronized with one another, but the time is not necessarily accurate with respect to UTC [5].

Clock synchronization aims to coordinate independent clocks. Even they are following the accurate time real clocks will slightly differ due their counting time. There are several problems associated with different clock rates [2].

## 2. Clock Synchronization in Wired and Wireless Communication

### 2.1. Wired Communication

Wired communication synchronization can be achieved through some algorithms and the following subsections discuss about those.

#### 2.1.1. Cristian's Algorithm

Cristian's algorithm is a passive method of time synchronization [1].Time server will have some source of maintaining a time like radio clock or some algorithms. Whenever a procedure call made to time server by some client its gives time as soon as possible [1], [2].

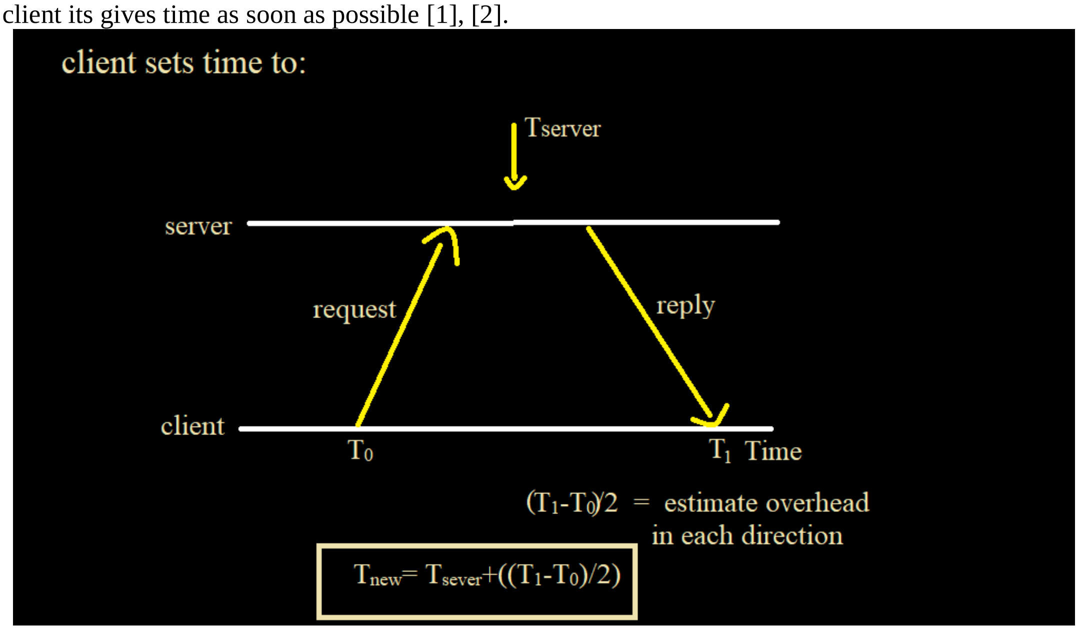



Fig: Time graph for Cristian's algorithm.

**2.1.2. Berkeley Algorithm**

Berkeley algorithm is used to overcome limitations of faulty clock and malicious interference in passive time server. Also it can overcome limitations of other active time servers [7]. This algorithm is used for the systems that have no global clocks or WWV Receiver. The time server will be in active state and asks time to time about what time other system or machines are following. Based on answers its computes an average time and tell to other machines to advance their clocks or minimize their clocks [1], [2].

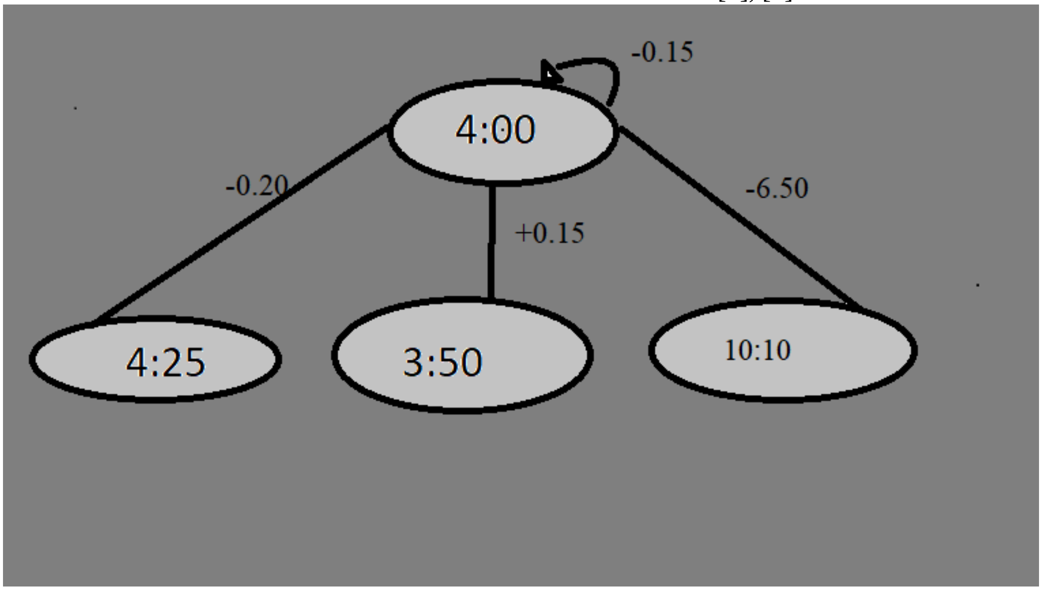
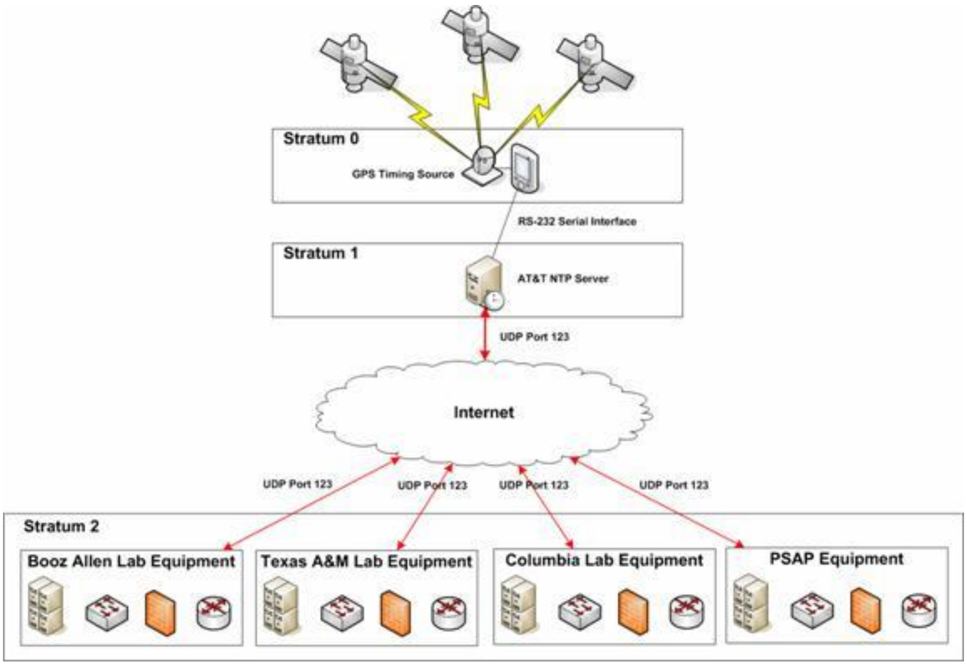



Fig: Example of Berkeley Algorithm

**2.1.3. Network Time Protocol**

This protocol is designed by Mills [6]. This protocol is usually a client-server model, but also works as peer-peer relations. Network time protocol is robust and oldest protocol that is used widely throughout the internet. NTP synchronizes connecting machines within a few milliseconds of coordinate universal time. It uses the intersection algorithm, a modified version of Marzillo's algorithm, to select accurate time servers [2]**.** The working will be as follows:

Some nodes are connected to GPS for synchronization. Stratum-L nodes serve as time servers are connected to stratum-L-1 nodes who will act as clients. When client send request along with time stamp, the upper level computers will work as servers and sends response by knowing and setting the information of global time [6].



**2.2. Wireless network synchronization:**

Design principles of clock synchronization for wireless sensor network:

The synchronization scheme should meet the following requirements on which the schemas are evaluated.

**1. Scalability:** A Synchronization scheme should complement with increasing number of nodes as in wireless sensor network any number of new nodes at any time is deployed in the network.

**2. Precision / Accuracy:** The need or purpose of accuracy or precision depends on for what purpose the sensor network is used. The amount of accuracy needed for synchronization will be based on application.

**3. Sensor nodes** are deployed at remote areas for some applications, if something goes wrong and sensor node fails then the recovery action is taken as soon as possible. This failure node should not affect other working nodes.

**4. Lifetime:** Synchronization among nodes should last for longer duration in some applications and some for shorter duration.

**5. Cost and Size:** Due to the advanced technology, nodes of wireless sensor are becoming small in size and cost decreasing. The synchronization method should be less in cost and size.

The protocol designed for synchronization should use less battery power as sensor node operates on battery power [6]. In wireless network time synchronization is very important for communication, where we should the below challenges as sensors are often placed in harsh environments:

- Fluctuations in temperature, pressure, humidity.
- Time synchronization solutions should be energy- efficient.
- Throughput variations, error rates, radio interferences,
- Topology changes, density changes, node failure.
- Limited processor speeds or memory (lightweight algorithms).
- Cost and size of synchronization hardware (GPS) [1], [4].

Synchronization problem consists of four parts as follows:

- Send time
- Access time
- Propagation time
- Receive time

Three current synchronization protocols are:

**2.2.1. Reference broadcast synchronization:**

This schema was proposed by Elson, Girod, and Estin. The send time and access time delays can be removed through RBC [6]. Many of the time synchronization protocols use 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(reference beacon) will broadcast a beacon to all the receivers. The beacon does not contain any timing information, instead the receivers nodes notes their local clocks and exchange information with neighboring nodes. The nodes will now able to calculate their offset. Thetiming is based on when the node receives the reference beacon.

RBS can be expanded from one broadcast to two receivers and synchronizes n- receivers, n is greater than two. We need to increase the broadcast with increase in the value of n [1], [2], [3].

**2.2.2. Timing-sync Protocol for Sensor Networks:**

TPSN is a traditional sender-receiver based synchronization that uses a tree to organize the network topology. The concept is broken up into two phases, the level discovery phase and the synchronization phase [1], [2], [3].

**Level discovery phase**

Level discovery phase creates a hierarchical topology of network in which each node is assigned a level. Root node first should be assigned. If one have's a GPS receiver the all others will formatted. If not one of the nodes should take an initiative and lead as root node. Once the root node is discovered then levels for all other nodes is to be assigned. Root node will at level zero for every time. It sends level-discovery packet to neighboring nodes consisting of its identity and level. Neighbors of root node will assign themselves as level-1 and send the level-discovery packet to their neighboring nodes and so on until all nodes receive some level. In this way we assign levels to nodes [3].

**Synchronization phase**

In synchronization phase again root node will take an initiation and broadcasts syn-packet to level-1 packets. These level-1 packets will wait for random time before initiating two-way messaging. The root node will send acknowledgment an level-1 nodes will adjust their clock according to root node. On hearing this communication level-2 nodes will also wait and follows the same process [3].

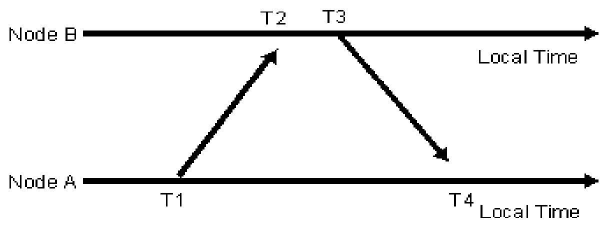



**Fig: C**ommunication between two nodes.

Figure illustrates the two-way messaging between a pair of nodes. This messaging can synchronize a pair of nodes by following this method. The times T1, T2, T3, and T4 are all measured times. Node A will send the synchronization-pulse packet at time T1 to Node B. This packet will contain Node A's level and the time T1 when it was sent. Node B will receive the packet at time T2. Time T3 is when Node B sends the acknowledgment-packet to Node A. That packet will contain the level number of Node B as well as times T1, T2, and T3. By knowing the drift, Node A can correct its clock and successfully synchronize to Node B. This is the basic communication for TPSN.

### 2.2.3. Flooding Time Synchronization Protocol

The FTSP protocol was designed by Miklos Moroti, Branisllarkusy and Simon and Akos [6]. This is similar to TPSN. The root node will transmit the time synchronization information with a single radio message to all participating receivers. The message contains the sender's time stamp of the global time at transmission. The receiver notes its local time when the message is received. Having both sender's transmission time and the reception time, the receiver can estimate the clock offset. The message is MAC layer time stamped, as in TPSN, on both the sending and receiving side. To keep high precision compensation for clock drift is needed [3].

FTSP was designed for large multi-hop networks. The root is elected dynamically and periodically reelected and is responsible for keeping the global time of the network. The receiving nodes will synchronize themselves to the root node and will organize in an ad hoc fashion to communicate the timing information amongst all nodes. The network structure is mesh type topology instead of a tree topology as in TPSN [6], [3].

## 3. Summary

Resource sharing is the vital objective of any distributed system. Proper synchronization should be mandatory in distributing the resources. Clock Synchronization through various algorithms has been described through this article. The need for clock measurement algorithm becomes necessary due to the increase of application of sensor networks either in star or mesh or any other framework. The article clearly presented the mechanisms, characteristics, and issues associated with those algorithms. This article may act as a path finder for other researchers in this domain.

In future work we may consider the performance analysis of various clock synchronization algorithms and propose some alternative.

## References:

[1]. Tanenbaw Andrews:Distributed Systems
[2]. Wikipedia: Clock Synchronization, Wired Communication and its Algorithms
[3]. Michael Roche: Wireless Network Communication and its Algorithms,2006;
[4]. Prakash Ranganathan, Kendall Nygard: Time Synchronization in Wireless Network Communication,2010;
[5]. Sami Rollins-Time Synchronization.
[6]. Devdatta Bagul, Prof. Suresh Kurumbanshi, Prof. Upendra Verma, “Survey on Clock Synchronization in WSN”,2013.
[7]. Neha N. Dalwadi1 and Dr. Mamta C. Padole2: “Comparative Study of Clock Synchronization Algorithms in Distributed System”, The Maharaja Sayajirao University, Gujarat, India