**Practical No. : 3**

**Aim :** To implement clock synchronization algorithm

**Problem Statement** : Design and implement a clock synchronization algorithm for prototype DS

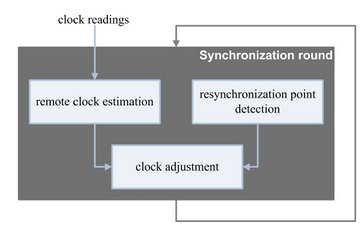
**Prerequisite :** Clock Synchronization concept

**Course Objective :** To learn communication methodology in distributed systems

**Course Outcome :** Able to learn and apply the concept of remote method invocation and Remote Procedure Calls

**Theory:**

Distributed System (DS) is a collection of computers connected via the high-speed communication network. Distributed system is one in which hardware and software components located on a network communicate and coordinate their actions only by message passing .



There are two types of Distributed Systems:

a) Homogeneous Distributed Systems (HDS): It is a distributed system such that all nodes have identical hardware, the same type of architecture and operating system.

b) Heterogeneous Distributed Systems (HeDS): It is a distributed system such that each node has their own operating system and machine architecture .

Each node in a distributed system can share their resources, e.g. the producer-consumer processes and the client-server processes, sharing of printer or scanner. But the resources can be limited therefore; they can be shared in either cooperative or competitive modes. Resources like a printer and scanner cannot be used by multiple processes simultaneously, so it must wait for one process to complete and then give chance to the next process.

Another example is like producer-consumer and client-server processes which work in cooperative mode . So there is a need of proper allocation of available resources, to preserve the state of resources and coordination between processes. To resolve these conflicts, clock synchronization is important. Clock synchronization can be implemented by using the physical local clock of each node.

Some of the following example shows the problems with unsynchronized clocks :

a) In a distributed banking system, if the timing and ordering of financial transactions are not tracked, it may raise inconsistent state in the system.

b) A distributed online reservation system in which the last available seat may get booked from multiple nodes if their local clocks are not synchronized.

c) There is a need to transmit a message from one node to another at any time. This will become difficult if sender and receiver clocks are not synchronized with each other.

Synchronization in DS can be achieved by using physical clock of the node. For synchronization purpose, each node in the system needs to share their local clock time with another node in the system. During this transaction (message passing of current clock time value) some factors like a communication link failure, fault tolerance, propagation time, non-receipt of acknowledgment, congestion in a network, the bandwidth of the communication link and routing mechanism affect and it may raise communication delay during this message passing which directly affects clock synchronization. In computers, clock synchronization performs based on the physical clock of a computer. Before going into detail about synchronization, first we need to understand how physical clock structure of computer is implemented and how it works.

Issues in Clock Synchronization A simple method of clock synchronization is that each node has to send a request message ‘time=?’ to the real-time server. The node gets a reply message with ‘time=t’. This method has following issues :

a) The ability of each node to read another node’s clock value. This can raise errors due to delay in message communication between nodes. Delay can be computed by computing the time needed to prepare, transmit and receive an empty message in the absence of transmission errors and system load.

b) Time must never run backward since it may lead to the repetition of events or transactions creating disorder in the system. Time running backward is just a perception, not actually it goes backward.

Reasons for Delay in Synchronization As discussed above, there are many reasons for a communication delay needs to be minimized to minimize delay and get nearby accurate time.

1. Communication Link Failure : For example, when sending a request message, communication link is working properly and message reaches to the server. If at the time of receiving message, communication link may fail due to some break. And client may not be able to get reply message. After recovery, reply reaches to the client which contains false time value.
2. Fault Tolerance : During message passing if any component fails, it may cause an inaccurate reading of clock time. So the system should be fault tolerant that can work in the faulty situation and minimize the clock drift value .
3. Propagation Time : Due to heavy traffic or congestion in the network, it may cause large propagation time from server to client. It may cause the inaccurate reading of the clock value in the reply.
4. Non Receipt of Acknowledgement : It may be possible that due to above reasons client will not get reply within a round trip time and therefore it sends multiple requests to server for synchronization.
5. The Bandwidth of Communication Link : Due to low bandwidth of communication link, congestion may occur in the network. Therefore request for time will not be able to reach the server or reply message will fail to reach client will affect clock synchronization.

CLOCK SYNCHRONIZATION ALGORITHMS

Clock synchronization is a method of synchronizing clock values of any two nodes in a distributed system with the use of external reference clock or internal clock value of the node. During the synchronization, many factors affect on a network. As discussed above, these factors need to be considered before correcting actual clock value.

Based on the approach, clock synchronization algorithms are divided as Centralized Clock Synchronization and Distributed Clock Synchronization Algorithm.

Algorithm :

Berkeley Algorithm This algorithm overcomes limitation of faulty clock and malicious interference in passive time server and also overcomes limitation of active time server algorithm. Time server periodically sends a request message ‘time=?’ to all nodes in the system. Each node sends back its time value to the time server. Time server has an idea of message propagation to each node and readjust the clock values in reply message based on it. Time server takes an average of other computer clock’s value including its own clock value and readjusts its own clock accordingly. It avoids reading from unreliable clocks. For readjustment, time server sends the factor by which other nodes require adjustment. The readjustment value can either be +ve or –ve. There are following limitations: i. Due to centralized system single point of failure may occur. ii. A single time-server may not be capable of serving all time requests from scalability point of view. The readjustment value can either be +ve or –ve.

Distributed Algorithms :

There is no centralized or reference time-server. It performs clock synchronization based on internal clock values of each node with the consideration of minimum clock skew value among clocks of different nodes in the system. Distributed clock synchronization algorithm overcomes issue of scalability and single point of failure as there is no common or global clock required. Processes make decisions based on local information and relevant information distributed across machines

**Berkeley’s Algorithm**

Berkeley’s Algorithm is a clock synchronization technique used in distributed systems. The algorithm assumes that each machine node in the network either doesn’t have an accurate time source or doesn’t possess an UTC server.

**Algorithm**  
1) An individual node is chosen as the master node from a pool nodes in the network. This node is the main node in the network which acts as a master and rest of the nodes act as slaves. Master node is chosen using a election process/leader election algorithm.

2) Master node periodically pings slaves nodes and fetches clock time at them using [Cristian’s algorithm](https://www.geeksforgeeks.org/cristians-algorithm/).

Diagram below illustrates how the master sends request to slave nodes.

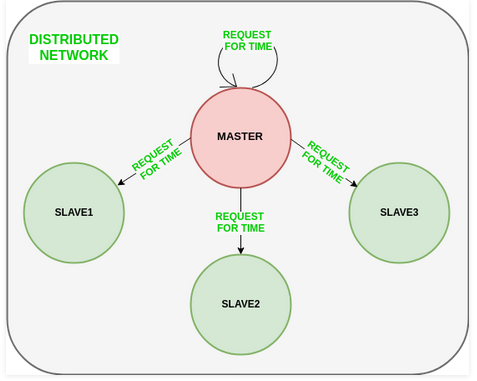
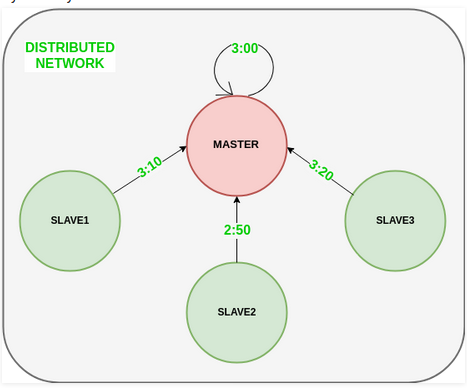
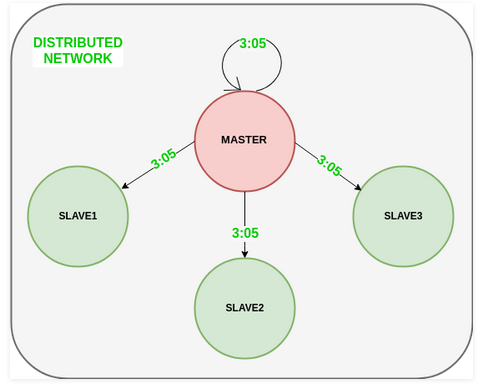


Diagram below illustrates how slave nodes send back time given by their system clock.



3) Master node calculates average time difference between all the clock times received and the clock time given by master’s system clock itself. This average time difference is added to the current time at master’s system clock and broadcasted over the network.

Diagram below illustrates the last step of Berkeley’s algorithm.



**Conclusion**

Thus, students understood Clock Synchronization Algorithm

**FAQ’s :**

What is Synchronization ?

What are various Clock Synchronization Algorithms?

Role of Clock Synchronization in Distributed System?

What are Two Types of Distributed Systems?