

Assignment No: 4 IS

Aim: Implement Berkeley algorithm for clock synchronization.

Objective: To understand Berkeley algorithm for clock synchronization & its implementation

Infrastructure: Python environment.

Software Requirements: Python 3.0

Theory:

• Berkeley algorithm:

Berkeley's algorithm is distributed algorithm for computing the correct time in a network of computers. The algorithm is designed to work in a network where clocks maybe running at slightly different rates & some computers may experience communication failures.

There are several variations of Berkeley's Algorithm

- Each computer starts with its own local time, & periodically send its time to master computer.
- The master computer receives timestamps from all computers in network.
- The master computer computes the average time of all the computers in the network.
- Each computer sets its clock to time it receives from master computer.
- The process is repeated periodically, so that over time, the clocks of all the computers in network will converge to

correct time.

Benefit: It is relatively simple to implement & under

Limitation: The time computed by algorithm is based on the network conditions and time of sending and receiving timestamps which if failed can cause the algorithm to stop working.

→ Scope of Improvement:

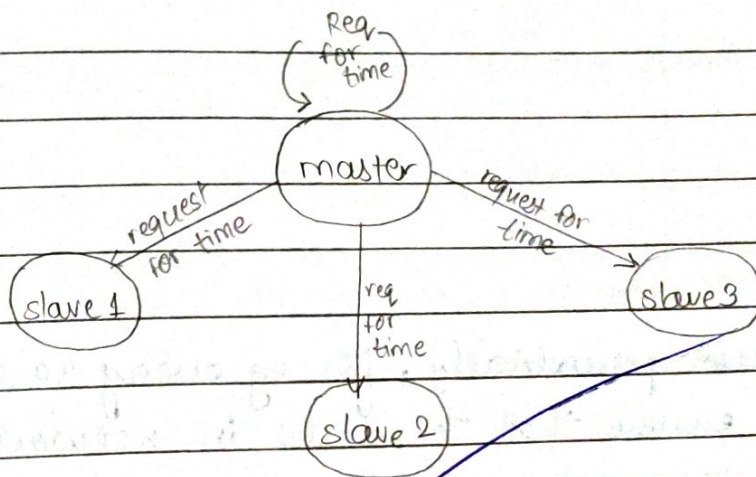
- Accuracy: The algorithm calculates the average time on timestamp received from all computers in network.
- Robustness: The algorithm requires a master computer which if it fails can cause algorithm to stop working.
- Synchronization: assumes that all clocks in a network are running at same rate, but in practice clocks may drift due to temperature, aging, other factors.
- Security: There is no security measures in algorithm to prevent malicious computers from tampering with timestamps they send to master computer which can skew the results of algorithm.

→ How to use Berkeley's Algorithm:

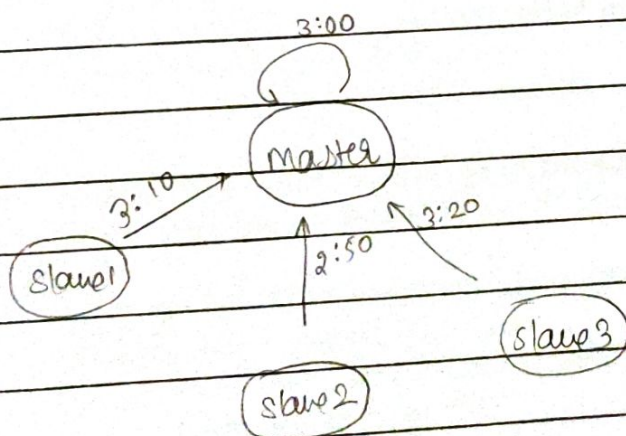
To use Berkeley's Algorithm, you would need to implement the algorithm on each computer in a network of computers. Here is a general overview:

Designate one computer in network as master computer. This computer will be responsible for receiving timestamps from all other computers.

on each computer set up a timer to periodically send the computer's local time to master computer.



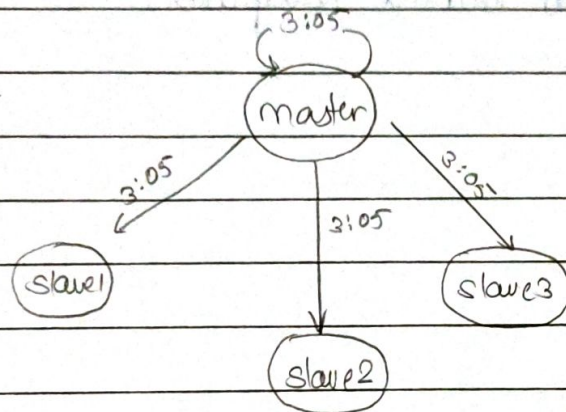
on the master computer, set up mechanism for receiving timestamps from all computers in network



on the master computer implement the logic for calculating average time based on received timestamps.

on the master computer, set up mechanism for sending calculated average time back all computers in network.

6. On each computer, set up a mechanism for receiving time from the master computer & setting computer clock to that time.



7. Repeat the process periodically, for eg every 30 seconds or 1 minute, to ensure that the clocks in networks stay synchronized.

→ conclusion: We learnt Berkeley's algorithm for clock synchronization & its implementation.