ASSIGNMENT NO. 5

Title: Clock Synchronization

Aim: To implement Berkeley algorithm for clock synchronization.

Objective:

1. To understand the basics of physical and Logical clock in DS.
2. To develop an n-node distributed system that implements Berkeley's time synchronization algorithm.

Related Theory:

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:

• 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.

• Master node periodically pings slaves nodes and fetches clock time at them using Cristian‟s algorithm.

• 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. Scope of Improvement

• Improvision in accuracy of cristian‟s algorithm.

• Ignoring significant outliers in calculation of average time difference

• In case master node fails/corrupts, a secondary leader must be ready/pre-chosen to take the place of the master node to reduce downtime caused due to master‟s unavailability.

• Instead of sending the synchronized time, master broadcasts relative inverse time difference, which leads to decrease in latency induced by traversal time in the network while time of calculation at slave node.

Scope of Improvement

- Improvision in accuracy of cristian‟s algorithm.

- Ignoring significant outliers in calculation of average time difference

- In case master node fails/corrupts, a secondary leader must be ready/pre-chosen to take the place of the master node to reduce downtime caused due to master‟s unavailability.

- Instead of sending the synchronized time, master broadcasts relative inverse time difference, which leads to decrease in latency induced by traversal time in the network while time of calculation at slave node.

Conclusion:

The Berkeley algorithm is a simple yet effective solution for clock synchronization in distributed systems. Its decentralized approach allows for resilience against failures, and it remains a relevant and widely used tool in ensuring accurate timekeeping between machines.

Outcome:

1. Students learn fundamental of clock synchronization in DS.
2. Students implemented Berkeley algorithm for clock synchronization.

FAQ:

1. What is difference between logical clock and physical clock?
2. Why is it necessary to synchronize the clocks in distributed real time system?
3. How the principle of Berkeley algorithm is used to synchronize time in distributed system?
4. What are other algorithms for clock synchronization in DS?