

CODE:

```
import java.io.*;
import java.net.*;
import java.util.*;

public class BerkeleyAlgorithm {

    // Define the port number that will be used for communication
    private static final int PORT = 1024;

    public static void main(String[] args) throws Exception {

        // Create a server socket to listen for incoming messages
        ServerSocket serverSocket = new ServerSocket(PORT);

        // Create a list to store the time differences for each node
        List<Long> timeDiffs = new ArrayList<Long>();

        // Create a new thread to handle the time requests from nodes
        Thread timeServerThread = new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try {
                        // Wait for a node to connect and request the current time
                        Socket clientSocket = serverSocket.accept();
                        ObjectInputStream in = new ObjectInputStream(clientSocket.getInputStream());

                        // Read the current time from the node's request
                        Date clientTime = (Date) in.readObject();

                        // Send the current time to the node as a response
                        ObjectOutputStream out = new
ObjectOutputStream(clientSocket.getOutputStream());
                        out.writeObject(new Date());

                        // Calculate the time difference between the server and the node
                        long timeDiff = (new Date().getTime() - clientTime.getTime()) / 2;
                        timeDiffs.add(timeDiff);

                        // Close the input/output streams and the socket
                        in.close();
                        out.close();
                        clientSocket.close();
                    } catch (Exception e) {
```

```

        e.printStackTrace();
    }
}
});
timeServerThread.start();

// Create a new thread to periodically send time requests to the server
Thread timeClientThread = new Thread(new Runnable() {
    public void run() {
        while (true) {
            try {
                // Connect to the server and send a time request
                Socket socket = new Socket("localhost", PORT);
                ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
                out.writeObject(new Date());

                // Read the current time from the server's response
                ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
                Date serverTime = (Date) in.readObject();

                // Calculate the time difference between the node and the server
                long timeDiff = (serverTime.getTime() - new Date().getTime()) / 2;
                timeDiffs.add(timeDiff);

                // Close the input/output streams and the socket
                in.close();
                out.close();
                socket.close();

                // Wait for a short period of time before sending the next time request
                Thread.sleep(1000);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
});
timeClientThread.start();

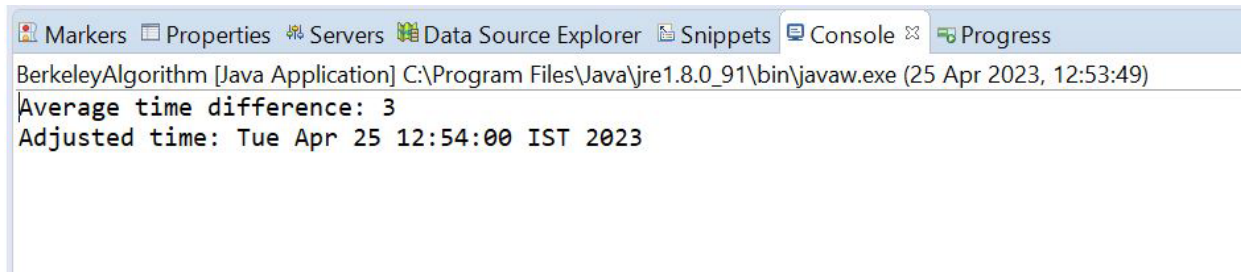
// Wait for a sufficient number of time differences to be recorded
Thread.sleep(10000);

// Compute the average time difference and adjust the node's clock
long sumTimeDiff = 0;

```

```
for (Long timeDiff : timeDiffs) {  
    sumTimeDiff += timeDiff;  
}  
long avgTimeDiff = sumTimeDiff / timeDiffs.size();  
System.out.println("Average time difference: " + avgTimeDiff);  
  
// Adjust the node's clock by adding the average time difference  
Calendar calendar = Calendar.getInstance();  
calendar.setTime(new Date());  
calendar.add(Calendar.MILLISECOND, (int) avgTimeDiff);  
System.out.println("Adjusted time: " + calendar.getTime());  
}  
}
```

OUTPUT:



The screenshot shows an IDE interface with a console window. The console title bar reads "BerkeleyAlgorithm [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (25 Apr 2023, 12:53:49)". The console output displays two lines: "Average time difference: 3" and "Adjusted time: Tue Apr 25 12:54:00 IST 2023". The IDE's top toolbar includes icons for Markers, Properties, Servers, Data Source Explorer, Snippets, Console, and Progress.

```
Markers Properties Servers Data Source Explorer Snippets Console Progress  
BerkeleyAlgorithm [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (25 Apr 2023, 12:53:49)  
Average time difference: 3  
Adjusted time: Tue Apr 25 12:54:00 IST 2023
```