LP 5

Name : Disha Chavan

Roll no. : 43212

Batch : P-10

Problem Statement : Implement Berkeley Algorithm for Clock Synchronization

In [3]:
```
# Python program imitating a clock server

# Importing necessary libraries

from functools import reduce
from dateutil import parser
import threading
import datetime
import socket
import time

# datastructure used to store client address and clock data
client_data = {}

'''
Nested thread function used to receive clock time
from a connected client
'''

def startReceivingClockTime(connector, address):

    while True:
        # Receive clock time
        clock_time_string = connector.recv(1024).decode()
        clock_time = parser.parse(clock_time_string)
        clock_time_diff = datetime.datetime.now() - clock_time

        client_data[address] = {
            "clock_time" : clock_time,
            "time_difference" : clock_time_diff,
            "connector" : connector
        }

        print("Client data updated with : " + str(address), end = "\n\n")
        time.sleep(5)


'''
Master thread function used to open portal for accepting clients over given port
'''

def startConnecting(master_server):
```

```
    # Fetch clock time at slaves/clients
    while True:
        # Accepting a client/slave clockc client
        master_slave_connector, addr = master_server.accept()
        slave_address = str(addr[0]) + ":" + str(addr[1])

        print(slave_address + " got connected successfully")

        current_thread = threading.Thread(target = startReceivingClockTime, args = (ma

        current_thread.start()


'''
Subroutine function used to fetch average clock difference
'''

def getAverageClockDiff():

    current_client_data = client_data.copy()

    time_difference_list = list(client['time_difference'] for client_addr, client in c

    sum_of_clock_difference = sum(time_difference_list, datetime.timedelta(0, 0))

    average_clock_difference = sum_of_clock_difference / len(client_data)

    return average_clock_difference


'''
Master sync thread function used to generate cycles of clock synchronization in the ne
'''

def synchronizeAllClocks():

    while True:

        print("New synchronization cycle started")
        print("Number of clients to be synchronized: " + str(len(client_data)))

        if len(client_data) > 0:
            average_clock_difference = getAverageClockDiff()
            for client_addr, client in client_data.items():
                try:
                    synchronized_time = datetime.datetime.now() + average_clock_differ

                    client['connector'].send(str(synchronized_time).encode())

                except Exception as e:
                    print("Something went wrong while sending synchronized time throug


        else:
            print("No client data. " + "Synchronization not applicable")

        print("\n\n")

        time.sleep(5)
```

```
'''
Function used to initiate the Clock Server / Master Node
'''

def initiateClockServer(port = 8080):
    master_server = socket.socket()
    master_server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    print("Socket at master node createdsuccessfully\n")

    master_server.bind(('', port))

    # Start listening to requests
    master_server.listen(10)
    print("Clock server started...\n")

    # Start making connections
    print("Starting to make connections...\n")
    master_thread = threading.Thread(target = startConnecting, args = (master_server,
    master_thread.start()

    # Start synchronization
    print("Starting synchronization parallelly...\n")
    sync_thread = threading.Thread(target = synchronizeAllClocks, args = ())
    sync_thread.start()



# Driver
if __name__== "__main__":
    #Trigger the Clock Server
    initiateClockServer(port = 8080)
```

```
New synchronization cycle started
Number of clients to be synchronized: 2
Socket at master node createdsuccessfully



Clock server started...

Starting to make connections...

Starting synchronization parallelly...

New synchronization cycle started
Number of clients to be synchronized: 0
No client data. Synchronization not applicable



Client data updated with : 127.0.0.1:62274

Client data updated with : 127.0.0.1:62273

New synchronization cycle started
Number of clients to be synchronized: 2



New synchronization cycle startedNew synchronization cycle started
Number of clients to be synchronized: 2

Number of clients to be synchronized: 2







Client data updated with : 127.0.0.1:62274

127.0.0.1:62290 got connected successfully
Client data updated with : 127.0.0.1:62290

Client data updated with : 127.0.0.1:62273

New synchronization cycle started
Number of clients to be synchronized: 3



New synchronization cycle startedNew synchronization cycle started
Number of clients to be synchronized: 3

Number of clients to be synchronized: 3
```

```
Client data updated with : 127.0.0.1:62274

Client data updated with : 127.0.0.1:62290

Client data updated with : 127.0.0.1:62273

New synchronization cycle started
Number of clients to be synchronized: 3



New synchronization cycle startedNew synchronization cycle started
Number of clients to be synchronized: 3

Number of clients to be synchronized: 3







Client data updated with : 127.0.0.1:62274

Client data updated with : 127.0.0.1:62290
```

In [ ]: