CSE 5306 Distributed Systems

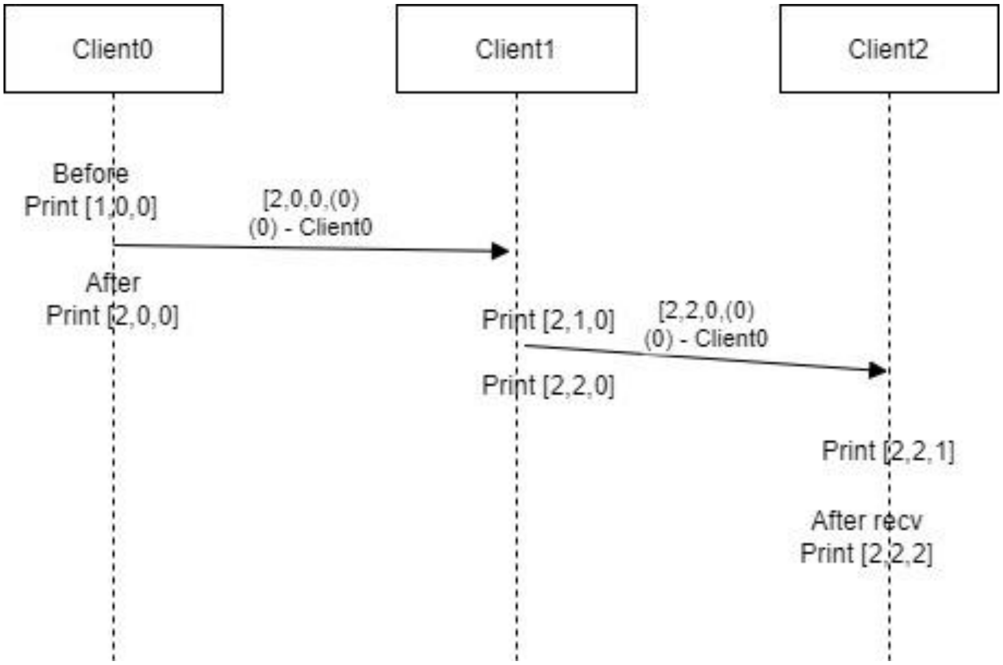
Suresh Kavadi – 1002040703
Ranjith Gopal Reddy - 1002028255

Vector Clock implementation on Distributed systems

Create 2 threads for each of the processes, 1 for sending data and other for receiving data. Communication between the nodes is happening socket programming and the solution is implemented in Python. Print the vector time before a process sends data and print the time after sending the data. The same happens at the receiving end as well.

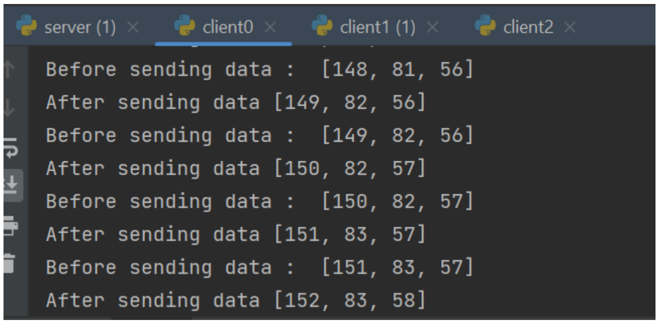
1. Initially all clocks are zero.
2. The local clock value for a node is incremented at least once before an event.
3. Each time a node sends a message, it increments its own timestamp vector.
4. Each time a node receives a message, it increments its own timestamp vector.

Below is the graphical representation of the implementation,



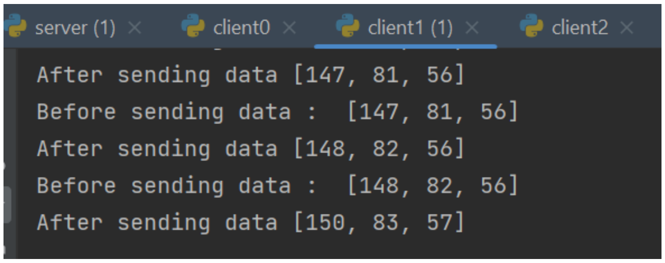
To make the results more understandable, Client0 sends messages every 5 seconds, Client1 sends every 10 seconds and Client2 sends after every 15seconds. And the same is communicated to all the nodes. Now let us look at the Terminal of Client0, CLient1, Client2.

Client0:



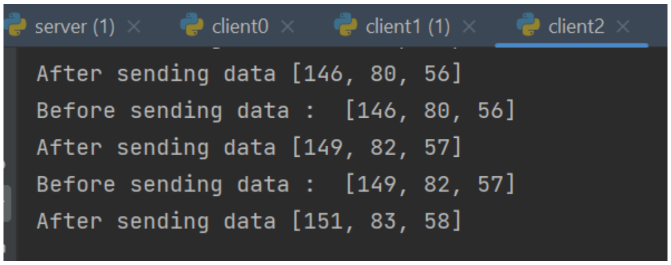
```
Before sending data :  [148, 81, 56]
After sending data [149, 82, 56]
Before sending data :  [149, 82, 56]
After sending data [150, 82, 57]
Before sending data :  [150, 82, 57]
After sending data [151, 83, 57]
Before sending data :  [151, 83, 57]
After sending data [152, 83, 58]
```

Client1:



```
After sending data [147, 81, 56]
Before sending data :  [147, 81, 56]
After sending data [148, 82, 56]
Before sending data :  [148, 82, 56]
After sending data [150, 83, 57]
```

Client2:



```
After sending data [146, 80, 56]
Before sending data :  [146, 80, 56]
After sending data [149, 82, 57]
Before sending data :  [149, 82, 57]
After sending data [151, 83, 58]
```

As you notice in the above outputs, the messages are exchanged between the nodes at the variably 5,10,15 seconds difference. Hence you can notice that the time at position 0 is higher than the remaining 2.

Execution details,

1. Go to the project directory and open terminal.
2. Run Server using command python server.py
3. Run Client using command python client<client_number>.py
4. Run all the 3 clients and notice the communication timestamps between the nodes.