**GROUP REPORT**

**PROJECT 2- Implementation of Asynchronous GHS algorithm**

Algorithm implementation:

We have used the Asynchronous GHS algorithm to build a minimum spanning tree.

**Message passing:**

1. Unlike the previous project, we used an asynchronous message passing between processes.
2. **Master:**
   1. Generate a pulse (starting with 1)
   2. Inform each process about the new pulse generated
   3. Wait until all processes reply “true”
   4. Generate the next pulse (go to step 2.a)
   5. Stop generating all pulses if all the processes have terminated

**Processes:**

1. Each time a process wants to send a message, it does the following:
   1. Generate a random number between 1-20
   2. Add the generated random number to the current pulse
   3. Schedule a message (add the message to the right channel) and attach with the timestamp generated in the previous step
   4. Continue with the next step
2. Each process does the following when it receives a new pulse from the Master
   1. Record the new pulse number
   2. Check the pulse number of first message of each channel
   3. If any of the scheduled messages pulse number is equal to the current pulse number, send the message to the destined neighbor
   4. Reply true to Master once all the scheduled messages have been sent

All the processes in the component are in synchronous, which means they are all doing the same step.

Each process does the following:

1. If you are the leader of the component:
   1. Initiate the broadcast to find MWOE
   2. Send test messages and find your local MWOE
   3. Wait for MWOE- converge cast from all the children
   4. Find the least MWOE and broadcast the same to the children
   5. Go to step **4**
2. If you are an intermediate node:
   1. Receive the find MWOE request from the parent
   2. Start finding MWOE and also forward the broadcast to the children.
   3. Wait for the MWOE converge cast from the children
   4. Find the least MWOE and send the same as a converge cast to the parent
   5. Wait for the connect broadcast from leader
   6. Go to step **4**
3. If you are a leaf:
   1. Receive the find MWOE request from the parent
   2. Start finding MWOE
   3. Send MWOE converge cast to your parent
   4. Wait for the connect broadcast from the leader
   5. Go to step **4**
4. Do the following:
   1. If your MWOE is same as the MWOE broadcasted by the leader
      1. Send a connect message to the process on the other component
      2. Wait for the reply
      3. If the reply is absorb
         1. Update the component details and level
         2. Broadcast the same to the processes of your old component
         3. Wait for the new leader which the connecting process will send
         4. Go to step **4B**
      4. If the reply is merge
         1. Identify who is the leader (the one with larger id)
         2. If you are the leader
            1. Make the process at the other end as your child
            2. Update the component details as (weight of core edge, vertex 1, vertex 2)
            3. Increment the level
            4. Broadcast the same to all the children
            5. Start next phase
         3. If you are not the leader
            1. Wait for the new leader broadcast message
            2. Set your old parent as your child
            3. Update you parent, component details and the level
            4. Broadcast the same to all children
   2. Else, do the following
      1. Wait for the new leader broadcast message
      2. Set your old parent to child
      3. Update parent, component and level
      4. Broadcast the same to your children
      5. If the connection with the other component is merge, start next phase. Else, wait for another new leader message

(Each time the level is modified, reply to the list of pending test messages)

1. How to reply test messages, B replies as below: (say, A sent a test message to B)
   1. If (B.Level >= A.Level)
      1. Compare component id’s
      2. If they are same, reply true
      3. Else, reply false
   2. If not, add this message to the list of the pending messages
2. How to handle connect messages: (say, A sent a connect message to B)
   1. If B has not found its MWOE yet, add this to the list of pending messages
   2. If MWOE is found
      1. Reply to pending messages
      2. Reply as below:
         1. If B.level > A.Level, reply absorb
         2. If they are same
            1. If your MWOE is with the same process, reply merge, else reply absorb
3. Termination happens when the leader broadcasts there are no “MWOE’s identified”