CS425 MP2

bmjain2 (Bhavana Jain) and dipayan2 (Dipayan Mukherjee)

We have implemented a distributed group membership using an extended ring topology. We list the components of the design:

- Finger Table: Helps us spread/disseminate the messages in $O(logN)$ time. Uses Chord like mechanism.
- Monitors: Nodes which track whether a certain node is alive by using heartbeat. Upon receiving a CRASH or LEAVE or JOIN message, the monitors are updated to reflect the changed state of the network.
- Children: Nodes which we track via heartbeat. This is also updated upon receiving a JOIN or LEAVE/CRASH message.
- Dissemination: Send the information to our neighbours and the entries in the finger table
- EventTable: checks the last updated timestamp received for a node, prevents duplication of messages in dissemination
- Membership Table : Maintains the complete network topology, along with every one's IP and the status of the node at present.

Implementation of node protocols for the following:

- JOIN : Send a request to an introducer, which assigns you an ID and sends your monitors to you, and disseminates your join request across the ring. Upon learning that a new node has been added other nodes, send the node your own information.
- CRASH: Monitors track your heartbeat, if no response received in two heartbeat, ask the suspect nodes neighbours, and upon receiving at-least one failure reply, set the suspect as crashed and disseminate in the network, and update your monitors. If no reply is received for your suspect message for 1 second, set it to CRASH and start disseminating. Upon receiving a CRASH message, set the node to dead, update your Monitors, and disseminate the information.
- LEAVE: Pressing `Ctrl + \` initiates the LEAVE process. The leaving node, disseminates its own LEAVE message, and the other node behaves similar to the CRASH message by setting the node as dead and updating its monitors.

Handling the INTRODUCER failure:

- When a node detects the introducer node has failed, it starts a process which periodically sends a special ping checking if the introducer it back or not, sending the introducer its own ID. Introducer upon rejoining receives the special ping, and disseminates its own information to all the nodes it knows, who further follow standard JOIN protocol. Hence this allows the introducer to re-learn the network.

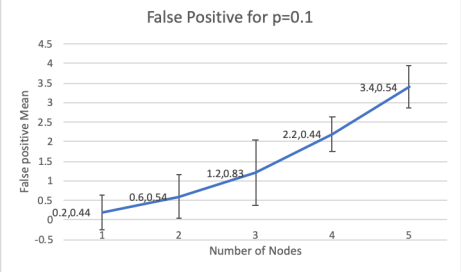
Background Functions :

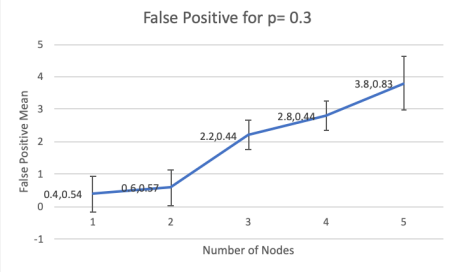
- Updating Finger Table : Checks periodically the network topology based on the Membership Table, and updates the values accordingly.
- Garbage ID collection : Run by introducer, maintains a list of dead node ID's which can be re-assigned to a node requesting to join the network. A dead node is added to the garbage list after 6 seconds of its failure to ensure every alive node is aware of the failure.

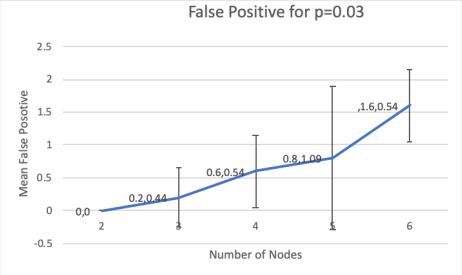
Marshalling : We have marshaled the message by converting each field to a string and appending it to a message string delimited by a comma (serialization). When this message is read at the listener port, it is deserialized by splitting on the delimiter and converting each field to its corresponding type.

Network Bandwidth Data: We used the `iftop` tool to measure the bandwidth usage.

a) Background bandwidth(heartbeat): The incoming background bandwidth is 46B/s for each child and outgoing bandwidth is 29B/s for each monitor.
b) Average bandwidth Usage :
    i) JOIN : 93 B/s
    ii) CRASH : 88 B/s
    iii) LEAVE : 20 B/s
c) False Positive for unreliable channel:







False positive : We see that the maximum number of failures is about n-2, as we can always detect the last two standing nodes, also with increasing unreliability the false positive rate increases as expected. Since the crash is detected by a fixed number of nodes, hence false positive/ number of nodes is a fairly constant value, and results in a fixed standard deviation of the error as the node size increase.