CS425 MP2

bmjain2 (Bhavana Jain) and dipayan2 (Dipayan Mukherjee)

We have implemented a distributed group membership using an extended ring topology.

Components of the design -

- Finger Table: Helps us disseminate the messages in $O(logN)$ time. Inspired from Chord DHT, distance between adjacent entries increases exponentially.
- Monitors: Each node heartbeats to one predecessor and two successors in the ring topology.
- Children: Set of nodes that send heartbeats to me. Store the timestamp of their last heartbeat to keep track.
- Dissemination: Send the message to monitors and entries in the Finger Table.
- EventTimeMap: stores the timestamp of the last message processed for a particular node, this prevents processing and disseminating the same message multiple times.
- Membership Table: Maintain a record of every node in the system. Each record contains virtual ID, IP and joining timestamp.

Event protocols:

- JOIN : Send a request to a fixed introducer, which assigns you a virtual ID on the ring, sends you your predecessor and successor entries (monitors), and disseminates your join request across the ring. When other nodes receive a JOIN message, they reply with their member record (their virtual id, ip, joining timestamp).
- CRASH: Monitors track your heartbeat, if no heartbeat is received in the last two heartbeat periods, ask the suspected nodes' monitors for status. If it is not removed from suspects list within 1 second, declare it failed and disseminate CRASH messages.
- LEAVE: Send a SIGQUIT signal (press `Ctrl + \`) to leave voluntarily. Disseminate LEAVE messages and exit.

Introducer failure handling:

When a node in the system detects the failure of the introducer, it starts a process which periodically sends a message containing its information (IP) to the introducer. When the introducer rejoins it disseminates a JOIN message and each alive node in the system replies with its information. This way the introducer relearns the network topology and membership information.

Background Functions:

- Update Finger Table: Update fingers periodically (every 10 seconds) based on the latest network topology.
- Garbage collection: The introducer maintains a set of ids of dead nodes in the system. These ids can be reassigned to new joinees to fill the gaps in the ring. A node is added to the garbage set at least 6 seconds after its 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.

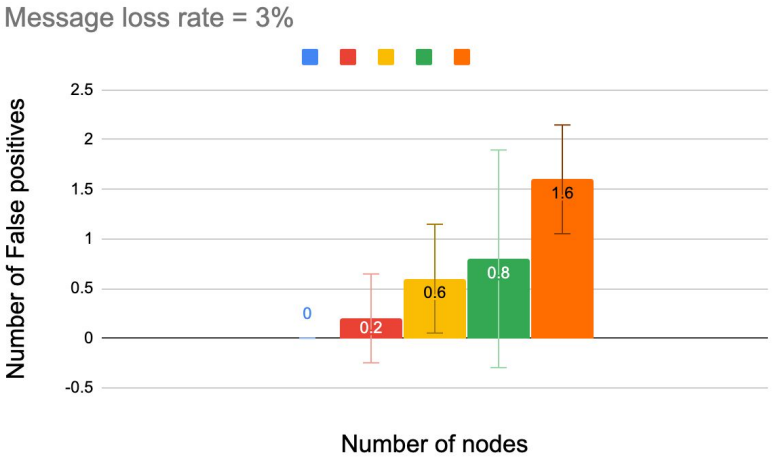
Debugging using MP1: We log all the major events - JOIN, CRASH, LEAVE, errors etc and store them in a logfile. We start the MP1 servers at each node in the system and perform a distributed grep using the MP1 client. For example, to check if all nodes received a CRASH message, we query for pattern=CRASH and check the output.

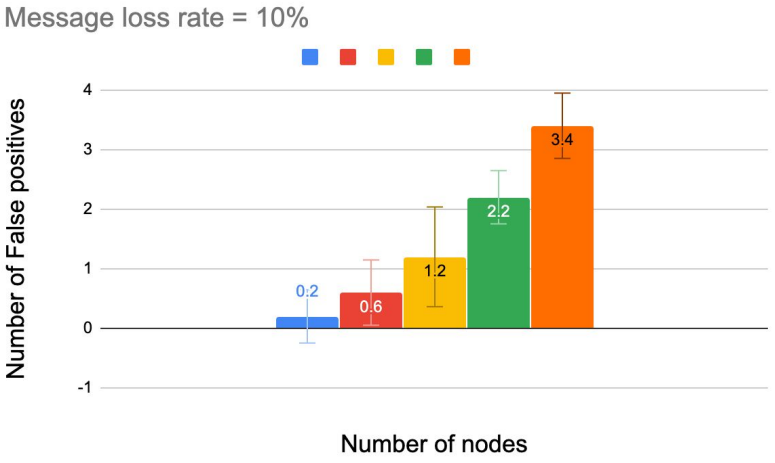
Network Bandwidth Data: We used the `iftop` tool to measure the bandwidth usage.

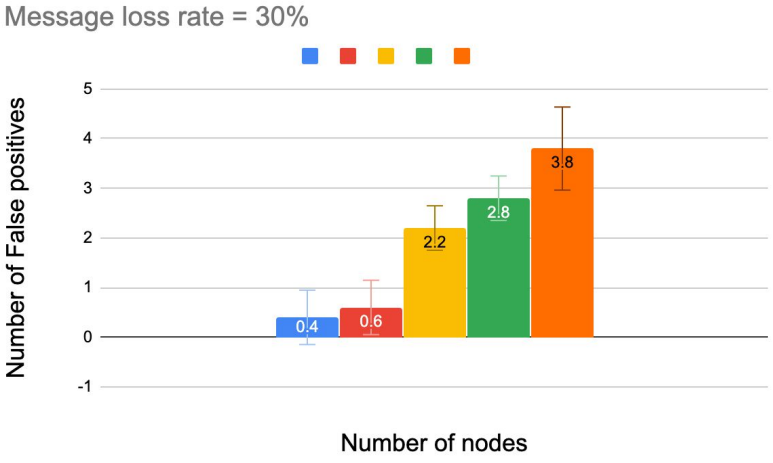
a) Background bandwidth (heartbeat): The incoming background bandwidth is 46 B/s for each child and outgoing bandwidth is 29 B/s for each monitor.

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:







When the total number of nodes in the system is low (n <= 3) the false positive rate is very low as there is an overlap in the predecessors and successors which overcomes the packet drop issue even for message loss rate = 30%. Once the size of the system is greater than 3, the false positive rate increases with the packet drop rate as expected. The standard deviation is high at lower loss rates of 3% and 10% because of the greater uncertainty in the event of packet drop.