Machine Programming 2 – Distributed Group Membership (lavisha2, vandana2)

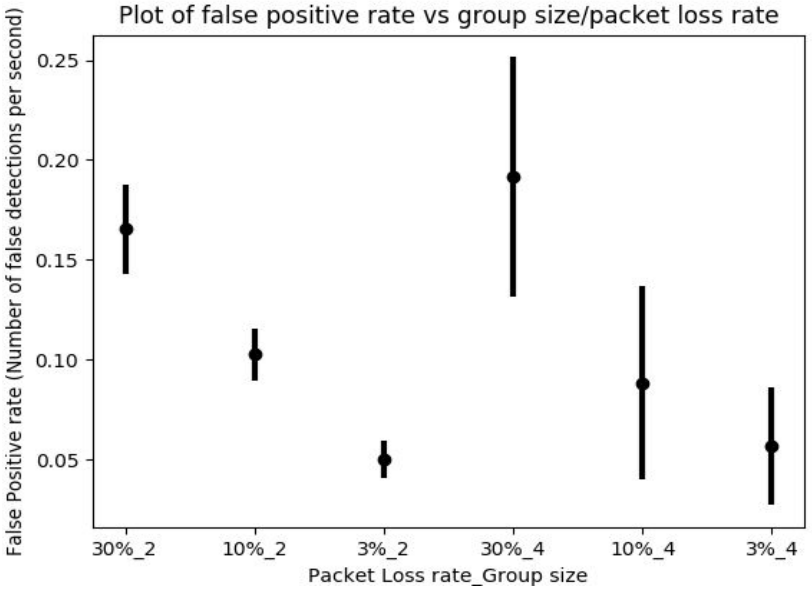
DESIGN: Our service maintains a membership list that contains information including nodeID, IPddress, Alive/Suspected/Failure/Leave field, Timestamp, Incarnation# of all other nodes part of the group. The membership list gets updated when a machine joins the group, voluntarily leaves the group or crashes. The nodes in the group are arranged in the form of a **virtual ring** and send/receive PING and ACK messages to two predecessors and two successors in the ring. When failure/node leave is detected it is piggybacked on top of PING messages and sent to **4 neighboring nodes**.

- This design is **robust to simultaneous failures** as each nodes is being tracked by (sent PINGS to) 4 other nodes. Hence, if 3 machines fail simultaneously, atleast 1 node in the group will detect all of their failure.
- Our design **scales to large N** since each node is communicating for both PING/ACK messags and failure dissemination with just a handful neighbors (4 nodes) hence reducing the overall number of messages in the group being constant.
- We have used **protobuf** to handle marshaled message format.
- A **new node** wanting to join the group, contacts the introducer. The introducer adds the new node's information to its membership list and sends back its updated list to the new node. The new node finds its neighbors in the virtual ring and starts sending PING messages to them. The introducer piggybacks the new node's information to its PING messages.
- **Failure detection:** In our service each node wakes up after a time period T, contacts its 4 neighbours in the virtual ring by sending a "PING" message and checks if it received an "ACK" response. If a neighbor hasn't responded with an "ACK" for a time period $T_{suspect}$, it is marked as suspect "S" in the membership list. Further if that neighbor still doesn't respond for another T_{fail} time period, that neighbor is marked as failed "F" in the node's membership list. Failures and Suspects are piggybacked with PINGS
- When a node receives an "ACK", it updates the neighbor's timestamp in its membership list and marks the node as Alive("A") if it was in suspect or fail mode.
- When a node receives some piggybacked information, to **update its membership** list, it does the following: Finds the information for that node ID in its list. If the incarnation numbers differ, keeps the latest version. If the incarnation numbers match, checks the timestamp and keeps the latest timestamp version. If the timestamp is the same, the F (Fail) > S(Suspect) > A(Alive) preference is followed.
- When a node wants to **leave the group**, before leaving it sends a special signal marking itself as "L" Leaving in the introducer's list. This info is also piggybacked.
- We have also handled **removing failed entries** from membership lists after a $T_{cleanup}$ time period. Further after this period, node Fail info is not piggybacked.
- When a node gets to know that it has been marked as Suspect ("S") or Failed ("F") due to packet loss, it increases its **incarnation number** in its list and piggybacks its own information with PINGS to its neighbors which further piggyback this info
- We have also logged the output into log files so that we can do a **grep using MP1**. We used some help from grep to debug and detect failures/suspects and node leaves.

BANDWIDTH

NORMAL: Usage for 4 machines with no membership changes
(i) Receiving bandwidth - 5410 B/s
Transmitting Bandwidth - 8420 B/s
CRASH: Usage for 4 machines when 1 node crashed
(ii) Receiving average bandwidth - 5800 B/s
Transmitting average bandwidth - 9290 B/s
LEAVE: Usage for 4 machines when 1 node voluntarily leaves the group
(ii) Receiving average bandwidth - 4130 B/s
Transmitting average bandwidth - 6775 B/s
JOINS: Usage for 4 machines when 1 node joins the group
(ii) Receiving average bandwidth - 6760 B/s
Transmitting average bandwidth - 11235 B/s
The bandwidth is almost comparable of the Normal case as compared to Crash, Leave and Joins. This is because each time we piggyback the entire membership list to all 4 neighbors. As expected in case of leave the average bandwidth reduces since a node leaves the group.

FALSE POSITIVE RATES



The dots represent the mean and the bars represent the standard deviations. The above plot shows the false positive rates for different packet loss rates (30%, 10% and 3%) for groups of 2 machines and 4 machines. Note that the above plot shows the number of falsely marked failures per second. As is expected the mean values of FPR decrease with decreasing packet losses in case of both 2 machines and 4 machines. For a packet loss rate of 3% the FPR is really low. The variance in the FPR is higher for a bigger group (4 machines) and higher for higher packet loss rate. A possible reason for this can be increased randomness because of more number of nodes. Comparing 2 nodes vs 4 nodes group, the mean values is slightly higher for 4 node groups in case of 30% and 3% packet loss.