Akil Karthikeyan Anandan (anandan3), Akshat Gupta (akshatg4)

MP2 Failure Detection Report

Design Overview

To implement the fail-stop model for the required protocols, we used IP + port + Timestamp (when node spawned) to uniquely identify each VM. We used configurable Tsuspect, Tfail, Tcleanup, and K values. T values are defined as X units, where the unit is also a configurable value. (For our experiments, we used Tsuspect = Tfail = Tcleanup = 5 units, K = 3, Unit = 25ms). We made VM1 the introducer, we did implicit join, where when a node has to join the group, it sends a join-req to VM1, and VM1 sends back a join-reply with its membership list. We also used UDP for all messages, marshalled, unmarshalled messages. Kept messages very compact and stored only what is necessary.

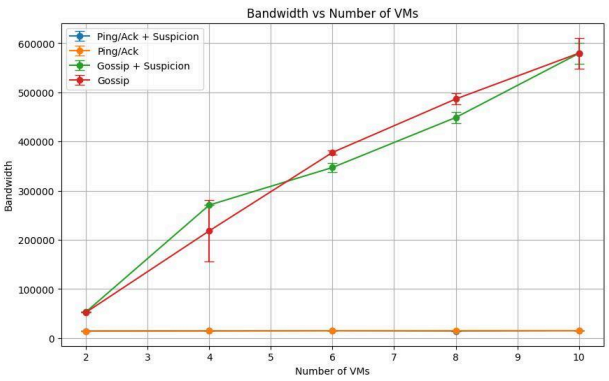
The state machines we used for the protocols are as follows:

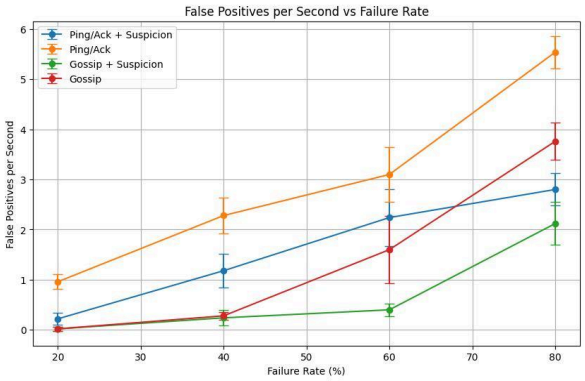
1. Ping/ACK + No Suspicion:
   Alive → (Send ping; Not ACKed / Not pinged by target for Tfail) → Delete (from membership list)
2. Ping/ACK + Suspicion:
   Alive → (Send ping; Not ACKed / Not pinged by target for Tsuspect) → Suspect
   Suspect → (Not ACKed / Not pinged by target for Tfail) → Delete (from membership list)
3. Gossip + No suspicion:
   Alive → (No alive gossip with higher heartbeat received from any other node for Tfail) → Failed
   Failed → (No alive gossip with higher heartbeat received from any other node for Tcleanup) → Delete (from membership list)
4. Gossip + Suspicion:
   Alive → (No alive gossip with higher heartbeat received from any other node for Tsuspect (or) receive suspect gossip with same or higher heartbeat) → Suspect
   Suspect → (No alive gossip with higher heartbeat received from any other node for Tfail) → Failed (Do not gossip about this node anymore)
   Failed → (No alive gossip with higher heartbeat received from any other node for Tcleanup) → Delete (from membership list)

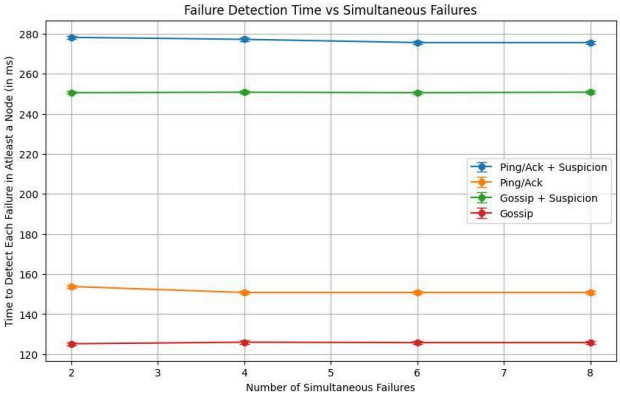
Worst case detection time for gossip (analyzing only for suspect, because no suspect is faster): Tsuspect + Tfail + Tcleanup = (5 + 5 + 5) * 25ms = 375ms (much lesser than 6s)

Worst case detection time for ping + suspect (we shuffle membership list and iterate through it for time bounded completeness): 2 * N - 1 VMs = 2 * 10 - 1 = 19 protocol periods, 1 protocol period = Tsuspect + Tfail = (5 + 5) * 25ms = 250ms => 19 * 250 = 4750ms (lesser than 6s)

Akil Karthikeyan Anandan (anandan3), Akshat Gupta (akshatg4)







Each point in these graphs are from 5 sample trials (mean and SD plotted)

a. Number of VMs vs. Bandwidth (Avg. for one node in Bytes/Sec - incoming + outgoing; averaged over 10s)
b. Failure rate vs. FPs/Sec (Avg. for one node; each FP is an actually alive node being deleted from membership list; averaged over 10s)
c. Sim. failures vs. Time to detect each failure in at least a node
d. Ping/Ack benefits more from suspicion mechanism, as seen from graph b. This is because gossip already filters out many False Positives (FPs) through redundant information spreading. Also, gossip without suspicion waits for Tfail + Tcleanup before deleting, whereas Ping without suspicion only waits for Tfail.
e. Failure detection time-wise (graph c) and FP rate wise (graph b), gossip without suspicion is better than ping/ack without suspicion. But, when it comes to bandwidth (graph a), ping/ack without suspicion is so much lighter. Failure detection time is lesser in gossip w/o suspicion compared to ping/ack w/o suspicion because in ping/ack each node is pinged individually and we wait for ack in every protocol period, but in gossip we just wait for timeouts simultaneously for all node (worst case analysis also explains this). FP rate is much higher in ping/ack w/o suspicion because it only waits for Tfail for an ack or ping from target node before deleting it, but gossip waits longer and also has redundant info spreading by all nodes, giving more chances for the alive node to be discovered. Bandwidth used by gossip w/o suspicion is much higher because of redundant info spreading whereas ping/ack is much lighter.