CS 425 Machine Problem 2 - Distributed Group Membership

Group 99 - Sanjay Pokkali (pokkali2) & Dhanush Suresh (dsuresh3)

Design:

Our implementation of the Distributed Group Membership system consists of 2 components, the introducer and the SWIM node where the introducer runs on one VM and the SWIM node runs on all the VMs. All the SWIM nodes are organized in a ring topology. Since we know only 3 nodes fail at any given time, we configure each machine such that it can only communicate with two of its predecessor nodes and two successor nodes. Additionally, in order to make sure the messages are platform independent, we dynamically generate an XML depending on the command that needs to be sent.

When the introducer starts, it opens a UDP port and listens for 3 commands: (i) *JOIN*: when a machine wants to join the group , (ii) *LEAVE*: when a machine wants to leave the group, and (iii) *FAIL*: When a node is detected to be failed. When any of these requests arrive, the introducer will update its local membership list and disseminate the updates to all the other members of the group. In addition to UDP packets as inputs, the introducer also accepts CLI commands to enable/disable suspicion and to update the message drop rate.

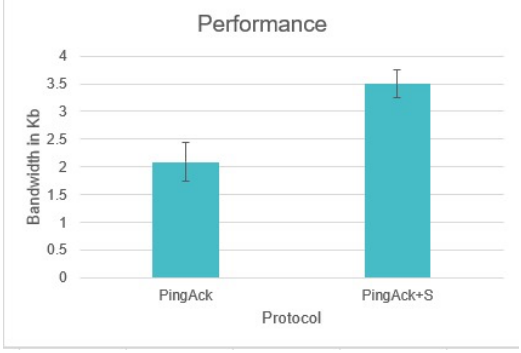
When a SWIM node starts, it will initially send a JOIN request to the introducer. Once the node joins the group and receives the membership list, it will start sending PINGs and ACKs to its 2 successors and 2 predecessors. In the PingAck model, if a node does not receive an ACK within 5 seconds, we mark it as failed and disseminate it to the introducer and the non failed successor and predecessors. In the suspicion enabled model, if a node is detected as suspicious, we will not mark it as failed but as SUS. This will be disseminated to the 2 successors and predecessors. If no communication is received from the suspected node, it is marked as failed.

Use of MP1:

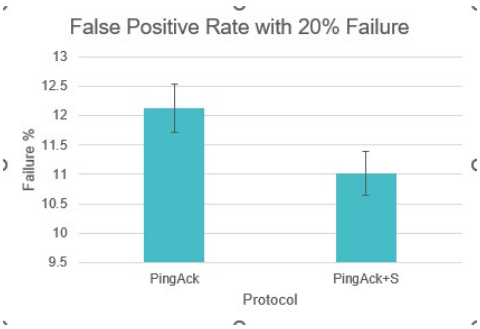
We will log PING, ACK, JOIN, LEAVE, FAIL and SUSPICION events in the log file during runtime. We used MP1 during development to grep useful information to debug issues.

Measurements:

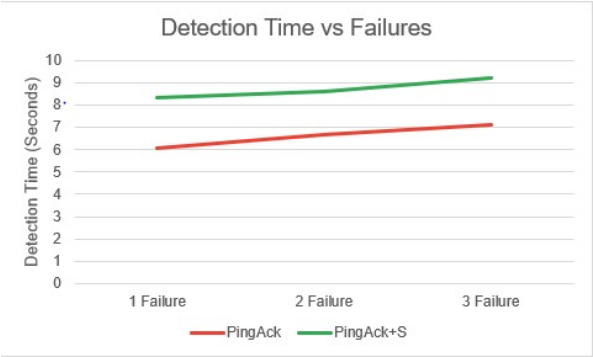
1.a)



1.b)

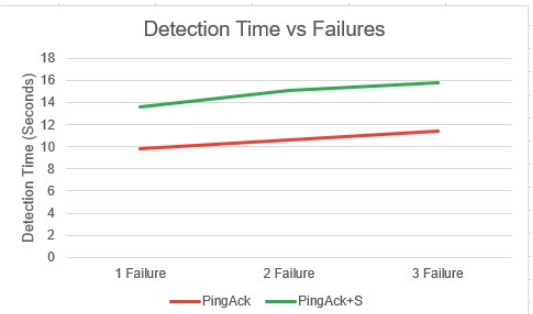


1.c)

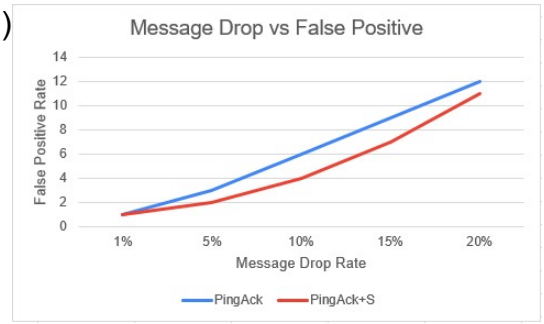


Analysis: The bandwidth of the PingAck with Suspicion is higher when compared to the PingAck without suspicion because of added suspicion message overhead. The false positive rate is higher for PingAck because the node is instantly deleted from the membership list. The failure time increases for the Suspicion protocol because of the added overhead with waiting. That is why the PingAck with Suspicion takes longer to detect failures than PingAck.

2.a)

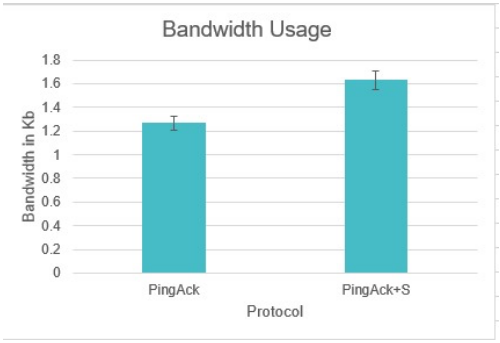


2.b)



2.c)

PingAck	PingAck+S	Difference
1.23	1.5	0.18
1.22	1.63	0.251533742
1.29	1.7	0.241176471
1.24	1.65	0.248484848
1.35	1.67	0.191616766



Analysis: Assuming we don't need to complete within the 10s protocol time, we modify the number of messages that are sent to induce a bandwidth of around 1.2Kb/s (This was measured using "iftop"). The detection time increases for failures because we are sending less number of Pings per unit time. The false positive rate is directly proportional to the message drop rate. The bandwidth is also decreased because of the reduced number of messages sent.

Code Repository Link