Saichetan Aitha (sa60)
Sonam Jain (sonyj9)

CS 425 MP2-G12: Distributed Group Membership

Overview

We extended our MP1 query system to implement a distributed group membership service. Each machine maintains a full and accurate membership list, pinging a randomly selected node and a subset of 4 predefined nodes every 5 seconds. If no acknowledgment is received within 1.5 seconds, the node is flagged and further action is taken based on the suspicion mechanism.

Introducer

We created an introducer machine to handle new machines joining the system. It listens for join requests, adds new nodes, and shares the updated membership list with all new machines that join. If the introducer is down, no new nodes can join, but existing processes continue as normal.

Ping-Ack

In the basic Ping-Ack process, if no acknowledgment is received, the node is immediately removed, and a failure message is broadcast to the system for quick detection. However, this can lead to inaccuracies, which are fixed by Ping-Ack+S.

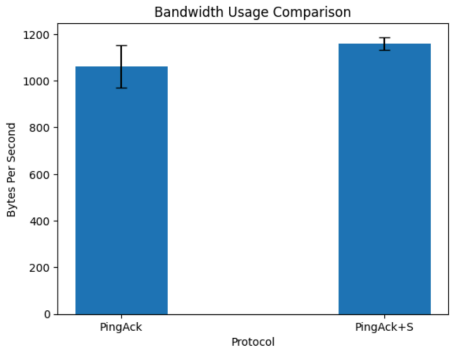
Ping-Ack+S

In the Ping-Ack+S implementation, a missed acknowledgment places the node in a "suspected" state for 8 seconds, instead of marking it as failed. All machines are notified and update their lists. The node increments its incarnation number during suspicion. If it responds within 8 seconds to any machine, its status is set back to "alive" with an updated incarnation number. Otherwise, it is marked as "failed" on all machines.

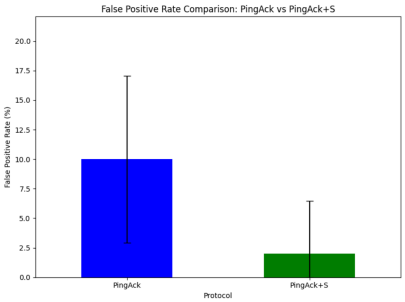
Explanation of Completeness and Accuracy for Ping-Ack+S:

Each machine is pinged by at least 4 others every 5 seconds, ensuring fast failure detection, even with up to 3 simultaneous failures. Failure information is broadcast immediately to all machines. The 1.5-second timeout quickly flags unresponsive nodes as suspicious, while the Ping-Ack+S mechanism waits 8 seconds before marking a node as failed, allowing time for recovery from network issues.

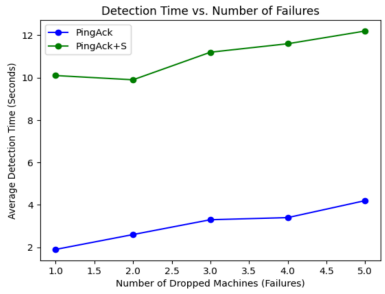
Ping-Ack vs Ping-Ack+S Analysis



PingAck+S results in slightly more bandwidth, which can be explained by any additional suspect messages being sent by machines in a no-failure scenario. All of the messages sent between machines are similar, besides more “suspect” messages.

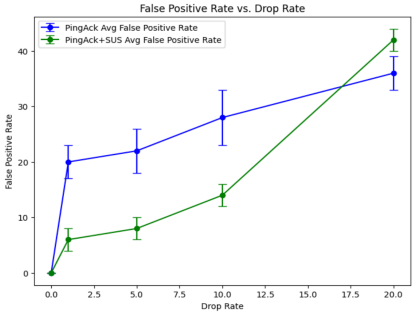


The false positive rate is much lower for PingAck+S, which is consistent with the addition of a suspicion mechanism. Nodes have a higher chance of recovering when a packet is dropped with PingAck+S. The standard deviations for both graphs were fairly high as the drop rate caused outlier results in a few trials.

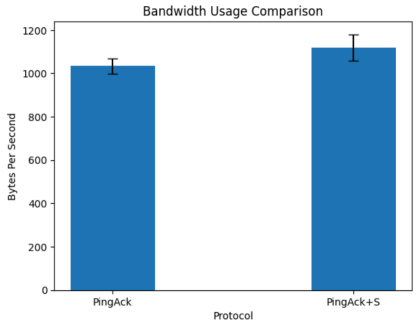


As the number of failures increases the time for detection slightly increases. This makes sense because there are fewer nodes present to detect failure. The time for detection is higher for PingAck+S because there are an additional 8 seconds given for PingAck+S to detect failure.

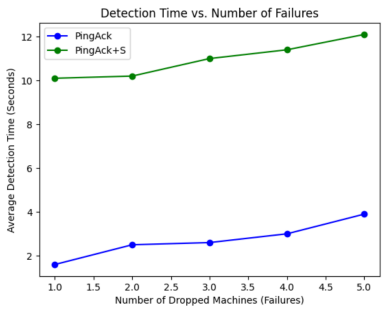
Results With Fixed Cap



As the drop rate increases the fall positive rate increases. This makes sense because more packets are being dropped, so more nodes are getting suspected. PingAck is typically higher because it immediately fails a node when a packet is dropped.



The bandwidth usage didn't change much with the bandwidth cap, as the scenarios were still the same.



The detection time with the fixed bandwidth cap did not change by much, but it was slightly lower. PingAck+S is still consistently higher than PingAck, which makes sense with the timeout value.