Distributed failure detector using All-to-All heartbeating and Gossip

Ricky (Ruiqi) Peng ruiqip Thomas (Qingyi) Huang qingyh2

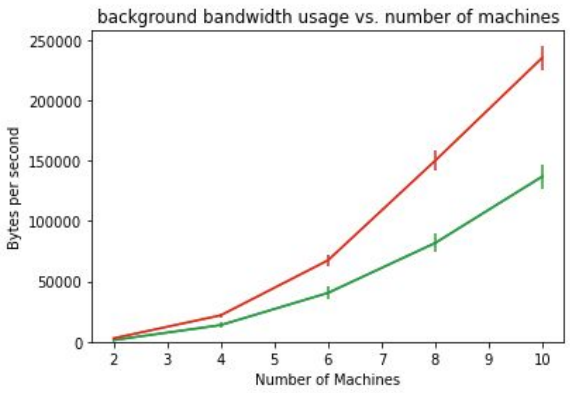

Figure 1

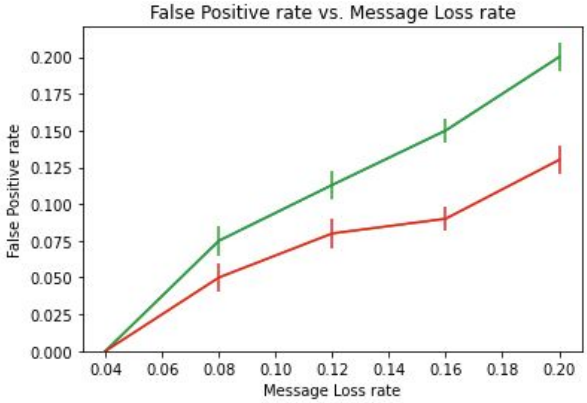

Figure 2

Red: All-to-all Green: Gossip

Figure 1: Bandwidth vs Number of Machines

We measured this by summing the length of marshalled bytes we use to send heartbeat messages in each machine in a given amount of time. We could clearly notice an increasing trend of bandwidth as the number of machines goes on; This is totally expected as we will only be sending heartbeats to running nodes. We could also observe that bandwidth occupied by all-to-all heartbeating are much higher than gossip's; This is due to the fact that we design our gossip method by randomly choosing a portion of running nodes (5 in this case) to epidemically spread the messages.

Figure 2: False-Positive rate vs Message loss rate

We measured this by having a conditional code wrapper around every sending behavior to manually control the message loss rate. The false positive rate of all-to-all heart beating generally matches the message loss rate. This is expected as the all-to-all heartbeating protocol requires every node to send messages to every other node, if all of the packets are successfully reached, then there wouldn't be any false positives ideally. However, the trend of gossip heartbeating slightly differs because since each node randomly chooses its recipients partially from every other node, it is expected to display a higher false positive rate as the randomness of the recipients' picks don't guarantee an on-time arrival of outcoming heartbeats.