MP2 Report

Arvind Arunasalam arunasa2
Justin Loew jloew2

Algorithm

Our algorithm is to send only heartbeats forward to the next node and messages about changes backwards. We only send messages about changes backwards. We only send to the next neighbor so that bandwidth is minimized. When scaled up to N, time to failure detection is scaled to O(N) but bandwidth is still conserved.

Out message format was ASCII. We convert our data to a string and then send it over the network. Our heartbeat is just our ID while our back propagation message is first divided to 3 groups which are joined, left and failed. Each group will have a list of the nodes and a timestamp. They are all converted into a string and sent over the network another reason that we converted our message to a string was that we were collecting garbage at the output. By converting out message to a string we could add some parity bytes to protect out information.

*

Mp1 was used for finding a specific error and when it happened. We generated log files for each change in the membership list and we stored it locally. To go through all the log files easily using a single node, the grep from mp1 was extremely helpful. We could immediately search up a change such as a node join, a node leaving or a node failure. We can also detect the difference in time between each failure by just putting the local time of each node alongside the change in the logs.

Bandwidth Calculation

For 4 nodes with no membership changes, the bandwidth is just (4 * 2 Bytes / 0.25) + ((4 * 10) Bytes/0.25) = 192 Bytes/sec.

(4 * 2 Bytes / 0.25) The first term is just the bandwidth of heartbeats sent each second.

4 for nodes, 2 for number of bytes in heartbeat, 4 times a second

((4 * 10) Bytes/0.25) This is the cost of the back propagation message

4 for nodes, 10 for just the basic message with messaging format, 4 times a second

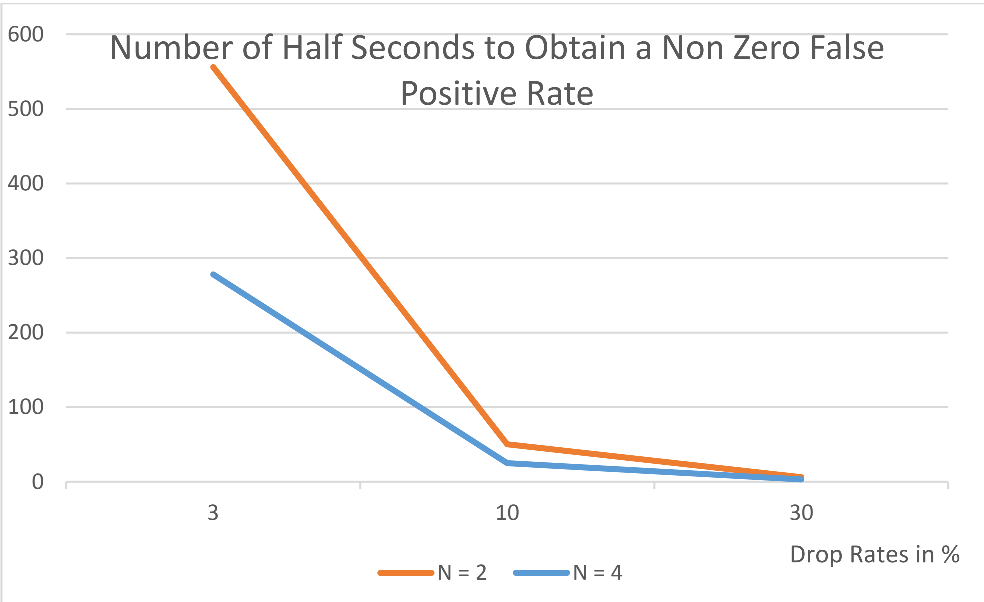
When a node leaved joins or changes, the bandwidth is increased by (4 * 5 / 0.5) Bytes per second

When a node leaves, the bandwidth is increased by (N * 7 – 4 * 2 – 4 * 10) / (0.25) Bytes per second
N is number of Nodes
7 is the increase in the back propagation message
-4 * 2 – 4 * 10 is the reduction in heartbeats generated and basic back propagation
4 times a second

When a node leaves, the bandwidth is increased by ((N+1) * 7 + 4 * 2 + 4 * 10) / (0.25) Bytes per second

N+1 is number of new Nodes
7 is the increase in the back propagation message
4 * 2 + 4 * 10 is the increase in heartbeats generated and basic back propagation
4 times a second

Graphs*



As we could not get our MP to work we had to calculate the probabilities of a node failing to receive a message due to the drops in the UDP connection.

As our send rate is 4 times a second and our receive rate is 2 times a second we will explain our calculations for a period of 0.5 seconds.

As we send our packets 4 times a second as expect to receive one 2 times a second we can afford to drop one packet every 0.5 seconds per path.
So a failure to detect a heartbeat or back propagation is $(\frac{\text{drop rate in \%}}{100})^2$. If a node fails to send a heartbeat, it is considered dead. So the probability for a miss of a heartbeat detection is $(\frac{\text{drop rate in \%}}{100})^2 * N$, where N is the number of nodes

The expected number of half-seconds for a Non Zero False Positive to occur is just
1 / $(\frac{\text{drop rate in \%}}{100})^2 * N$.

This is how we calculated the numbers for our graph.