**COP5615 Project 2 Bonus Report: Gossip Algorithm**

**Group Members:**

* Venkata Ramana Pamarthy, UFID: 76176980
* Sreeram Aditya Potukuchi, UFID: 31395861

**Gossip Failure Model Implementation:**

* We have successfully implemented failure model, where we induce failure into the gossip algorithms by taking an additional parameter in the program execution called *Fault.*

*c(main\_bonus).*

*c(gossip\_bonus).*

*c(pushsum\_bonus).*

* To execute, run the following commands

main\_bonus:start(arg1, arg2, arg3, arg4, arg5)

where,

arg1 = number of nodes

arg2 = topology (one of *line*, *2d*, *imp3d*, *full*)

arg3 = algorithm (either *gossip* or *pushsum*)

arg4 = gossip count (number of messages each actor receives before converging)

arg5 = fault count (number of nodes to kill)

* Then, we are killing the Fault number of nodes to induce failures to check how the 4 topologies converge in gossip.
* Then the algorithm runs as usual and at the end, we are sending a message to our program for the count of number of nodes that failed to converge using this command:

*final\_convergence\_name ! {ok,Count}*

* This gives the output of the number of nodes that failed to converge as follows:

Text

Description automatically generated

* It is clear from the above graph that, as we increase the failure percent(x-axis), the number of nodes converging(y-axis) decreasing, meaning that there has been a break in the chain of neighbors causing a break in the communication.

**Pushsum Failure Model Implementation:**

* We have successfully implemented failure model, where we induce failure into the pushsum algorithms by taking an additional parameter in the program execution called *Fault.*

*c(main\_bonus).*

*c(gossip\_bonus).*

* To execute, run the following commands

main\_bonus:start(arg1, arg2, arg3, arg4, arg5)

where,

arg1 = number of nodes

arg2 = topology (one of *line*, *2d*, *imp3d*, *full*)

arg3 = algorithm ( *pushsum*)

arg4 = gossip count (0 for pushsum)

arg5 = fault count (number of nodes to kill)

* Then, we are killing the Fault number of nodes to induce failures to check how the 4 topologies converge in pushsum.
* Then the algorithm runs as usual and at the end, we are sending a message to our program for the count of number of nodes that failed to converge using this command:

*final\_convergence\_name ! {ok,Count}*

* This gives the output of the number of nodes that failed to converge as follows:

Text

Description automatically generated

* We then plot a graph based on the converged nodes we get from the output above and it looks like below:
* It is clear from the above graph that, as we increase the failure percent(x-axis), the number of nodes converging(y-axis) decreasing, meaning that there has been a break in the chain of neighbors causing a break in the communication.