

COP 5615 Distributed Operating Systems Principles Project 2

Implementation-

- We have implemented the Gossip algorithm and PushSum algorithm.
- Gossip Algorithm-A **gossip protocol** is a procedure or process of computer peer-to-peer communication. Some distributed systems use peer-to-peer gossip to ensure that data is disseminated to all members of a group. Some ad-hoc networks have no central registry and the only way to spread common data is to rely on each member to pass it along to their neighbours.
- So we have three functions- start, begin gossip and continue gossip.
- When we pass the no. of nodes from the console we spawn those many actors, each actor has these three functions individually. Moving on start is creating the neighbour mapping depending upon the respective topology assigned for each file.
- Then **BeginGossip** just initiates the gossip for every actor or node in the network mapping across every node to **ContinueGossip** that propagates the message on the network
- **ContinueGossip** has two major functionalities. First it's the **iteration** variable that is like a timer which keeps on emitting the rumour after a specified interval regularly and next is finding a random neighbour for that particular actor node and then propagating it to the next respective node randomly in the network so these two things are done under **ContinueGossip**.
- Then we check **count** that is the no. of rumours received by one particular node, if it exceeds 10.
- **For Pushsum algorithm-** we start the protocol by passing (0.0) and (0.0) for weights for random node.
- The start function first creates the neighbour array of each actor node depending upon the topology that is assigned. Further, the second function continue gossip computes the actual computation of receiving the income sum and weight adding it to the existing income sum and weight and forwarding the average of the values received to the next random node and for the convergence the function checks if the difference between ratio of s/w against the new ratio of s/w is less than 10^{-10} for three consecutive rounds and if it is less than 10^{-10} then the node converges.
- Termination-The actor terminates when the actor's ratio of s/w does not change more than 10^{-10} even after three consecutive rounds.
- Good topologies for gossip protocol are torus, 2D, line and 3D for large numbers of actors. For smaller values of actors Full network also gives good convergence values. The reason behind this is based on the underlying principle of gossip protocol of spreading rumors.
- Line-

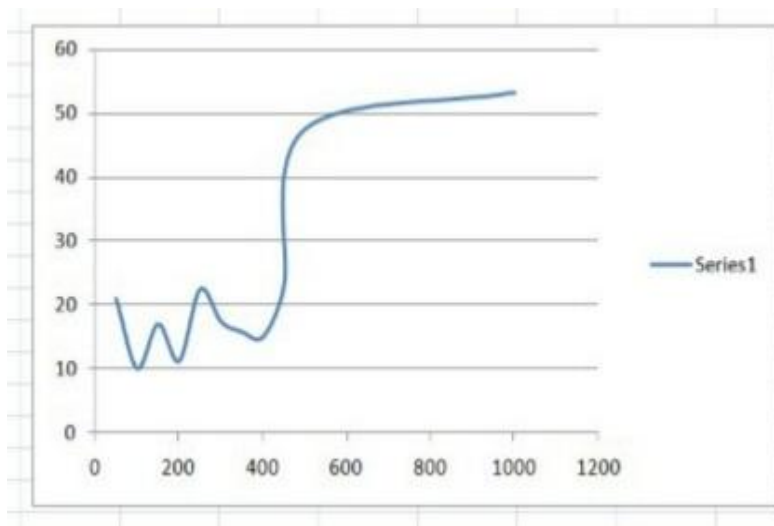
1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

In **line topology** nodes have either one or two neighbours not more than that the first and last node as shown in the figure have only one neighbour each towards its right and left respectively whereas other nodes have two neighbours each as they are in between the first and last node.

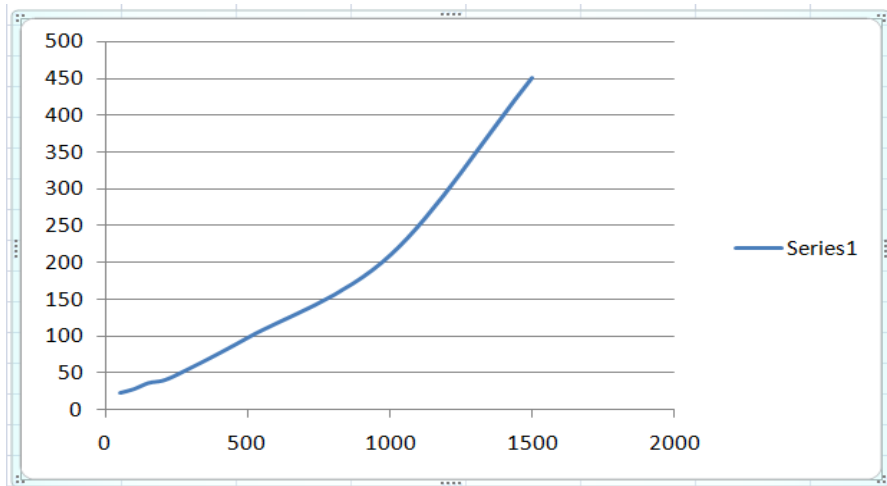
- Full-In Full network all actors are neighbors of the actor to which the rumor is spread first thus the program terminates quickly as every other actor will have the rumor from the first node but for larger numbers of actors this topology fails as the process overhead of asynchronous calling among actors dominates the advantage of having all nodes as neighbors.
- 3D Grid- As the name suggests this is a grid in 3 dimensional space so there can be many cases of each node having different amount of neighbour nodes. We consider superimposing 2D grids one on top of the other.
Cases-
 - Imperfect 3D- Every node is connected to a random node other than the desired neighbours in the 3D grid.

Results-

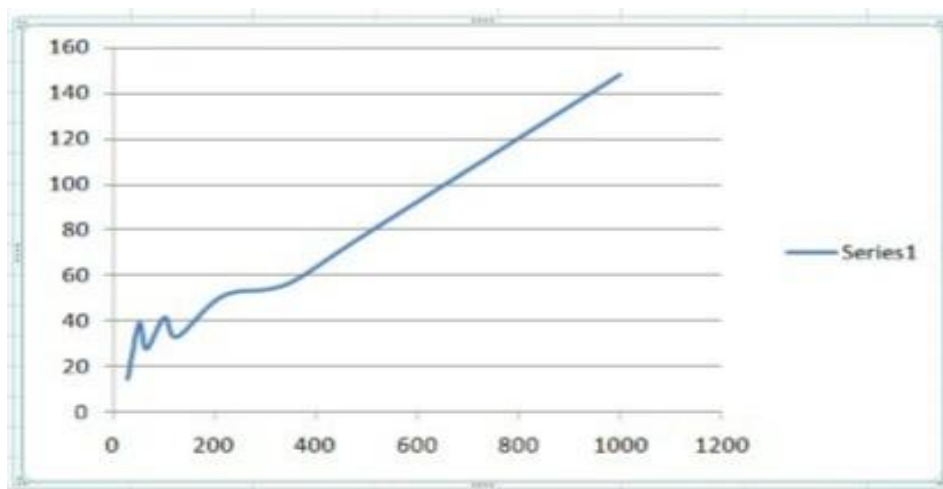
- **Gossip Full-**



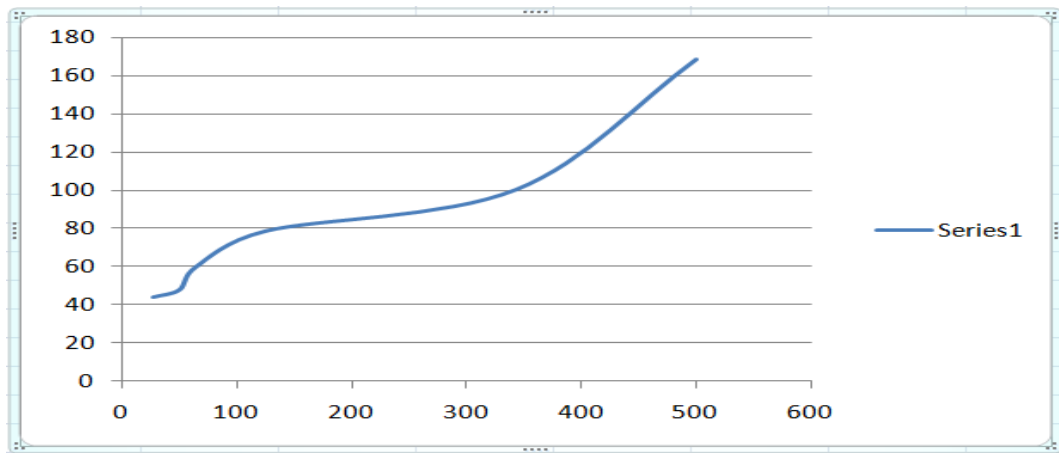
- **PushSum Full**



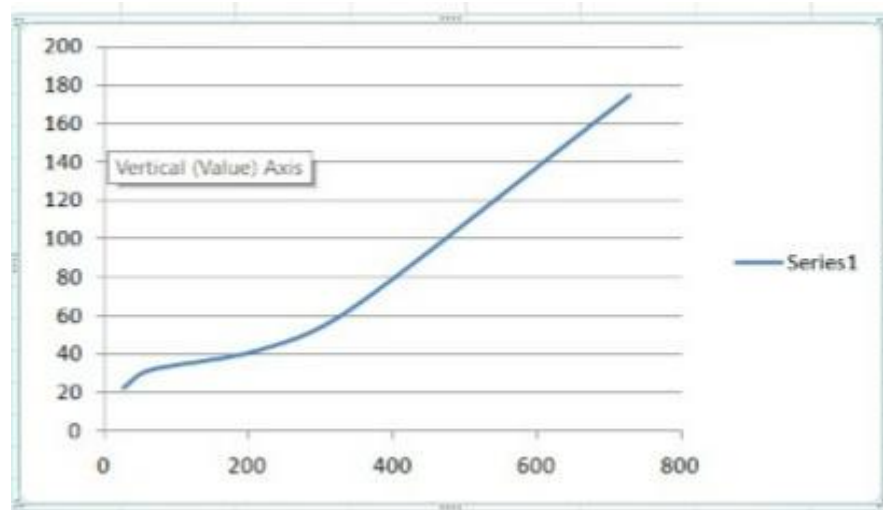
- **Gossip 3D**



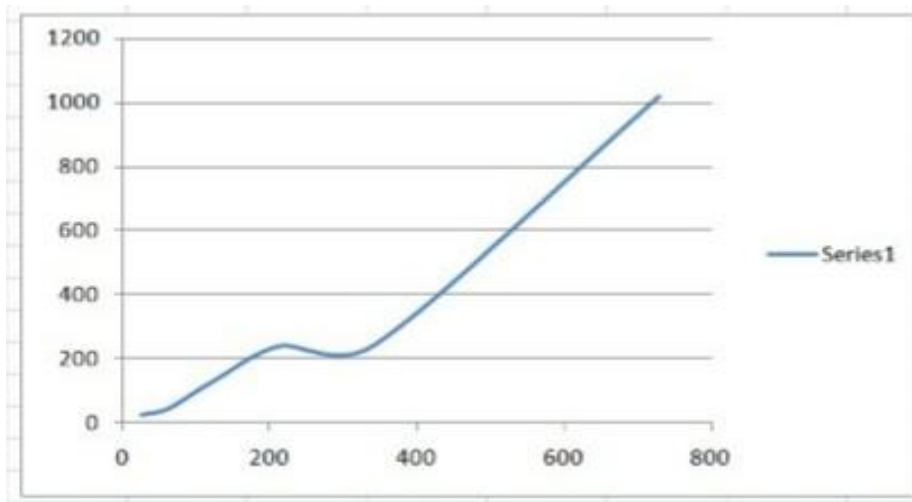
- **Pushsum 3D**



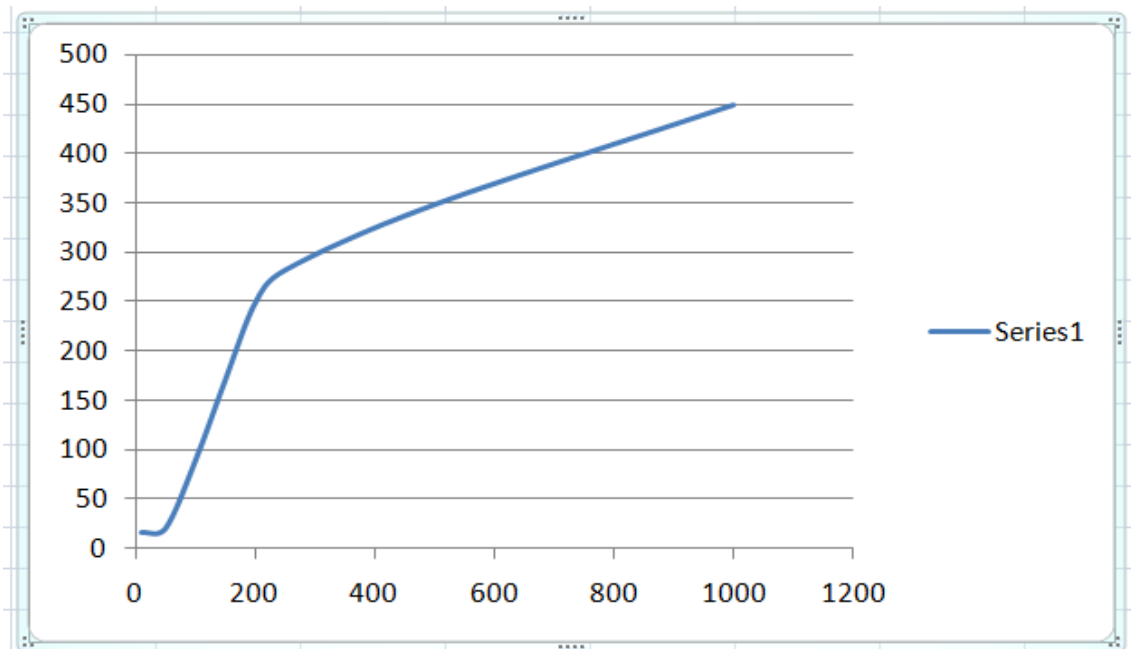
- **Gossip Imperfect 3D**



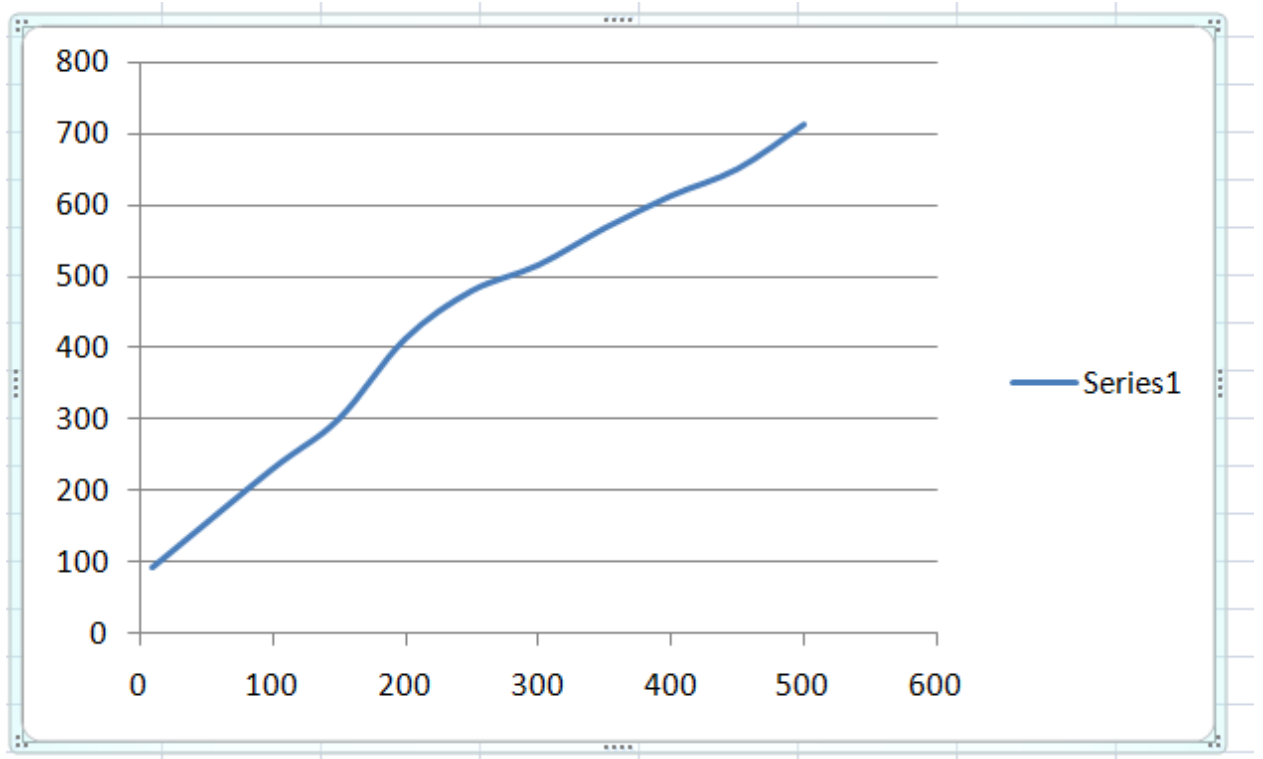
- **Pushsum Imperfect 3D**



- **Gossip Line**



- **Pushsum Line**



Note-

The algorithm that we have built converges successfully but takes a long time for gossip algorithm using full network topology.

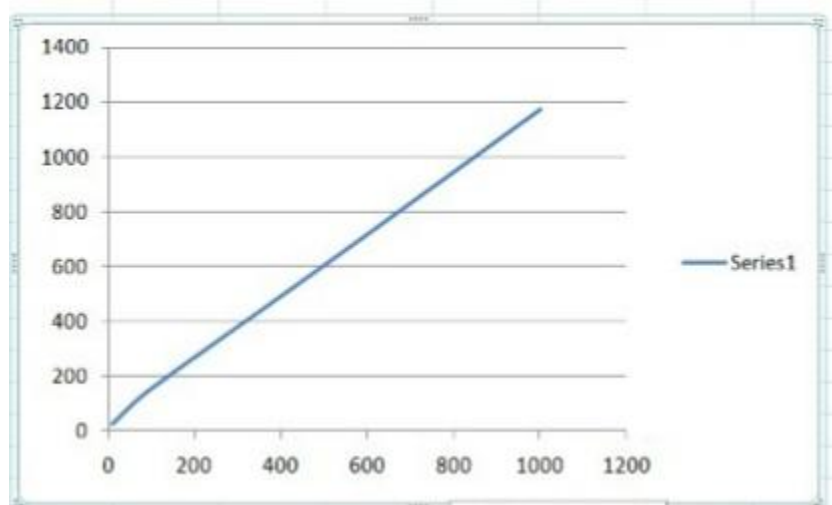
Bonus-

- **3D Grid**-If there is 1 percent failure 3D works fine but if there is 5 percent failure it gets stuck beyond that it does not converge.
- **Line**- From our observation, When there is a node failure, the nodes cannot travel to their adjacent neighbors in case of failed nodes, the line topology won't be able to converge for the case of larger nodes. It fails beyond 50 for our case.
- **3D Imperfect**-As there are more neighbours for 3D imperfect since every node other than the assigned 3D neighbor has one extra random node as a neighbor therefore the convergence time for 3D imperfect is slightly more than 3D. The convergence fails over 10% node failure.

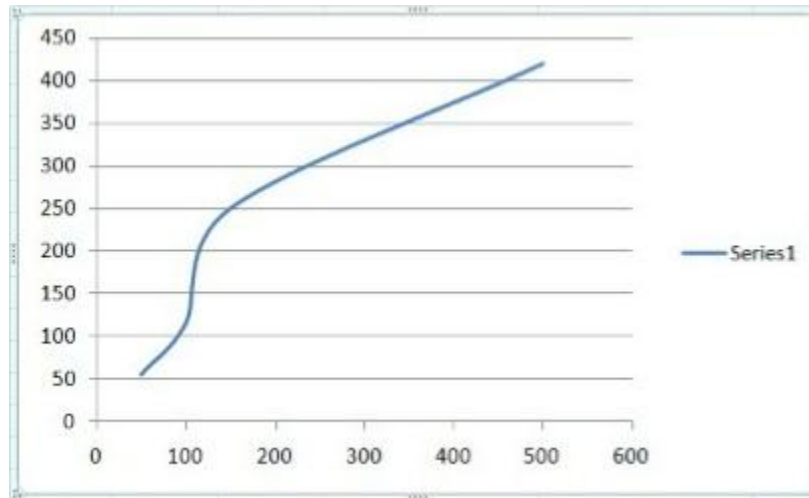
- **Full Topology**-In the case of full topology the nodes converge but the robustness of the nodes converging is affected greatly; the following graph shows the observations with 10% node failure. The time greatly increased with more percent of node failure.

Results-

- **Gossip Full-**



- **Pushsum Full-**



- **Gossip Line**

