DOS Project 2

Implementation of Gossip Algorithms

Team members:

Akshay Rechintala (4581-6988)

Venkatalakshmisupraja Keerthikaushal Gudur (8241-4961)

Aim:

The aim of this project is to implement the Gossip Algorithm and the Push-Sum protocol.

Implementation details:

Gossip: Each actor selects another actor at random and passes the message to it. Once it receives the message 10 times, it becomes dormant. The list of neighbors for every actor is updated when an actor checks its neighbors list for a random node and gets to know that this node is dead. If the neighbors list for an actor becomes empty, the actor becomes dormant in this case also.

The purpose of Gossip protocol is to disperse a message in a without centralized authority. So the time it takes for the rumor to be heard by every node defines the efficiency of the protocol. Hence convergence occurs when every actor has received/ heard the message at least once. An "ets" table is created to maintain the count of list of nodes that heard the rumor at least once. Once the count value reaches the number of nodes, the algorithm converges.

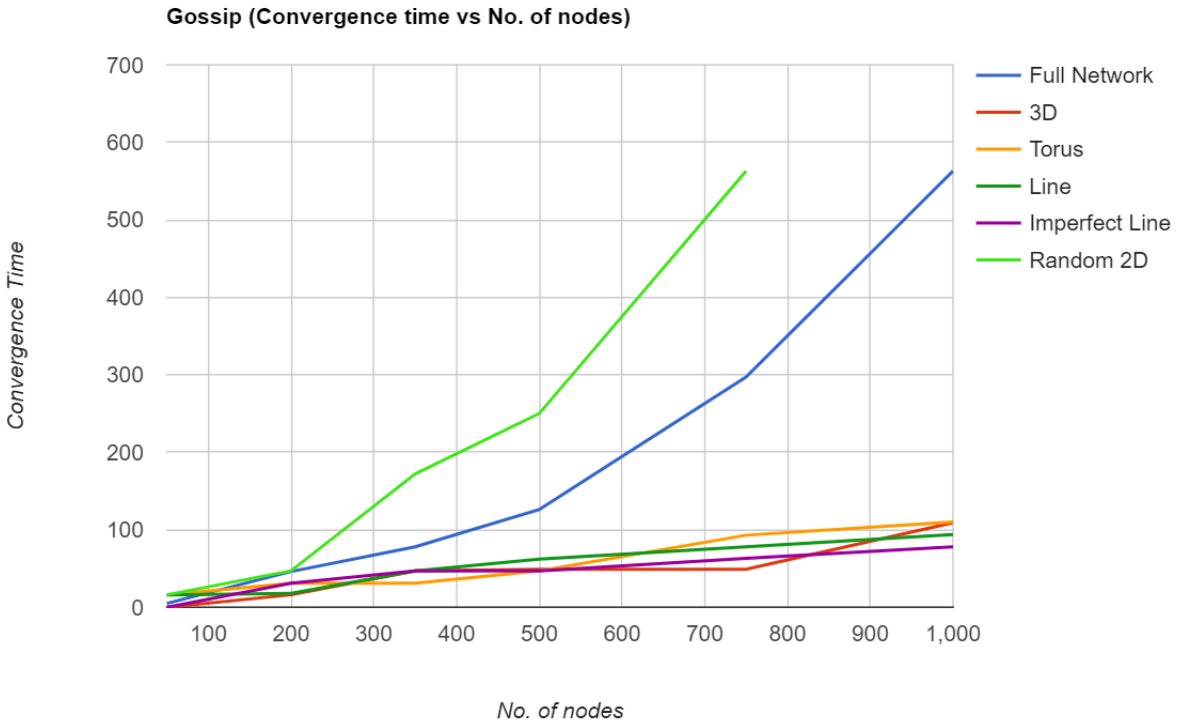
Push Sum: Every actor selects another actor at random and sends it s/2 and w/2 values. It updates its s and w values as s/2 and w/2. The actor converges when the change in its s/w value is less than 10^-10 for 3 times. When all the actors reach convergence, the algorithm converges. Each node receives the message 3 times to ensure fault tolerance of the network.

The following are the observations for convergence times across all topologies for **Gossip Protocol**:

Observations: (see graph below)

- Imperfect Line performs the best of all
- However, as the number of nodes increases, 3D performs better than all other topologies.
- This is closely followed by Torus. Between 400 to 600, the performance of torus has steadily improved.

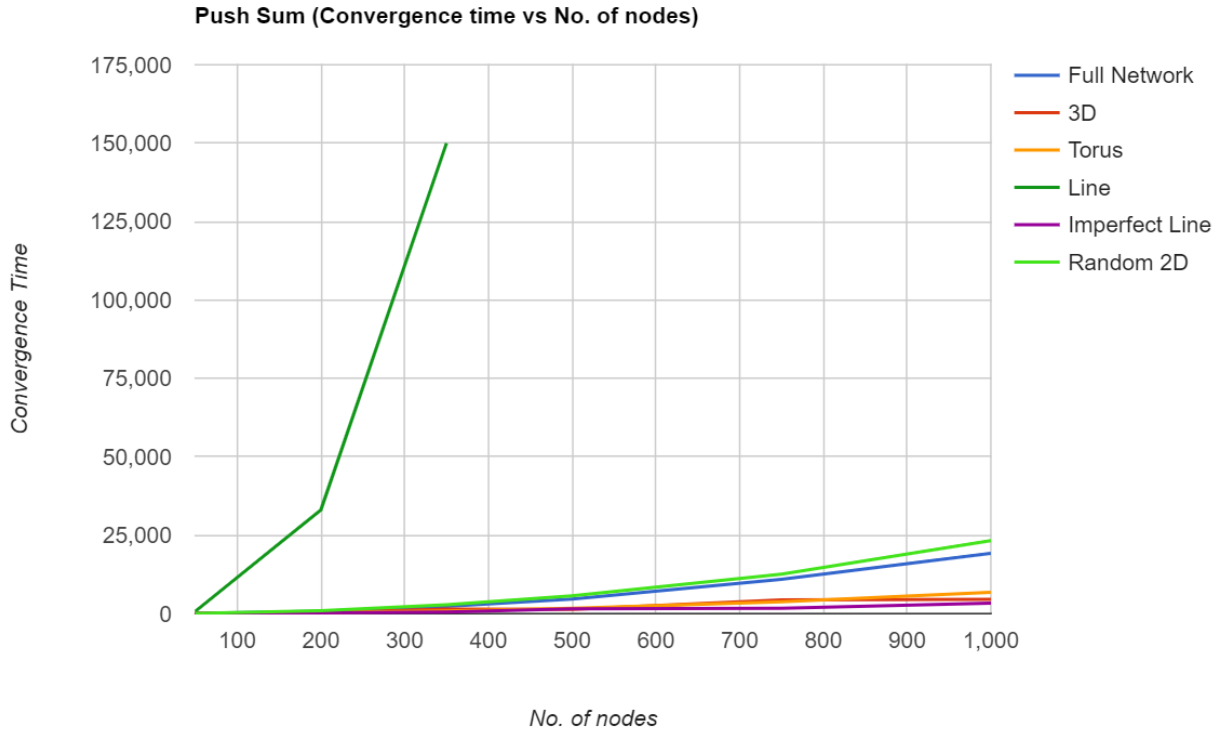
- Full Network and Random 2D surprisingly do not better inspite of the randomness. Instead topologies that constrainedly allow randomness such as the imperfect line perform better.



The following is the convergence time vs number of nodes graph for **Push-Sum** protocol:

Observations: (see graph below)

- Line performs the worst of all topologies
- Random 2D and Full network perform almost the same
- Torus, 3D and Imperfect line perform better.
- Imperfect line performs the best of all topologies.
- Next is 3D followed by Torus.
- Upto around 350 nodes, all topologies except line perform the same way.



The maximum possible network capacity for each topology for the given constraints is as follows:

Gossip:

Topology	**Maximum nodes**	**Time for Convergence**
Imperfect Line	10,000	36250
Torus	15000	6859
3D	10,000	1922
Random 2D	5000	60859
Full network	5000	39219
Line		

Push Sum:

Topology	Maximum nodes	Time for Convergence
Imperfect Line	7000	19234
Torus	5000	196234
3D	5000	105750
Random 2D	2000	109906
Full network	5000	549203
Line	750	1259214