

DOS PROJECT-2

GOSSIP ALGORITHM & PUSH-SUM IMPLEMENTATION IN ELIXIR

Ratna Prakirana (UFID : 3663-9969)

Eesh Kant Joshi (UFID : 1010-1069)

Brief Description

The aim of the project is to implement Gossip and Push-Sum algorithm in Elixir for different network topologies like Line, Imperfect Line, Rand2D, 3D grid, Torus and Full Network and analyze their convergence time.

Network Topologies

Network topology plays a vital role for the convergence time of the Gossip/Push-Sum algorithm. The topology determines who can be the neighbor of the current node. In this project we are considering six topologies namely like Line, Imperfect Line, Rand2D, 3D grid, Torus and Full Network.

- **Line:** Actors are arranged in a line. Each actor has only 2 neighbors - one left and one right, unless it is the first or last actor. For the first and the last actor, the number of neighbors is 1.
- **Imperfect Line:** Same as Line topology but each node selects one more random neighbor from the list of all actors.
- **Random 2D Grid:** Actors are positioned on the grid and they are connected randomly to other actors. Each actor has 4 neighbors, unless it is on corner, which has 2 neighbors or on edge, which has 3 neighbors
- **3D Grid:** Actors form a 3D grid. Each actor has 6 neighbors, unless it is on corner, which has 3 neighbors, on edge, which has 4 neighbors or on surface, which has 5 neighbors.
- **Torus:** Actors are arranged in a 2D grid. That is, each actor has 4 neighbors (similar to the 2D grid) but both directions are closed to form circles.
- **Full Network:** Every actor is a neighbor of all other actors. That is, every actor can talk directly to every other actor.

Gossip algorithm

Gossip algorithm is an information propagation algorithm designed for distributed systems. The aim of the algorithm is to send the message to every node in the network with the help of random neighbor selection.

The Gossip protocol works by initiating the process from a single actor which forwards the message to the other actors. The point of convergence is reached when actors listens to the message at the most 10 times (arbitrary threshold).

Push-Sum algorithm

Push-Sum is a variant of Aggregate calculation algorithm designed to calculate sum/average quantity for a distributed network. The algorithm works as follows:

- The Push Sum algorithm works by sending messages in the form of pairs(s, w) where s is the value of the actor number and $w = 1$ for each actor.
- The propagation converges when the s/w ration doesn't change when compared to a predefined value. (In our case 10^{-10}) for three consecutive times.

Convergence Model for our algorithm

1. Gossip Protocol

- Nodes which have received a message 10 times are maintained in a list (passive_list).
- Every node which reaches that limit is placed in the list by "Supervisor" Node.
- A non-working or passive node cannot send any messages and participate in the gossip.
- The following algorithm converges if a set number of nodes are placed in the passive list according to the topology selected and the main process is killed.
- If 20% of the nodes out of the total number of nodes are non-working for **Line** topology, convergence is reached.
- If 40% of the nodes out of the total number of nodes are non-working for **Imperfect Line** topology, convergence is reached.
- If 40% of the nodes out of the total number of nodes are non-working for **Random2D** topology, convergence is reached.
- If 10% of the nodes out of the total number of nodes are non-working for **3D grid** topology, convergence is reached.
- If 10% of the nodes out of the total number of nodes are non-working for **Torus** topology, convergence is reached.
- If 70% of the nodes out of the total number of nodes are non-working for **Full network** topology, convergence is reached.

2. Push Sum Algorithm

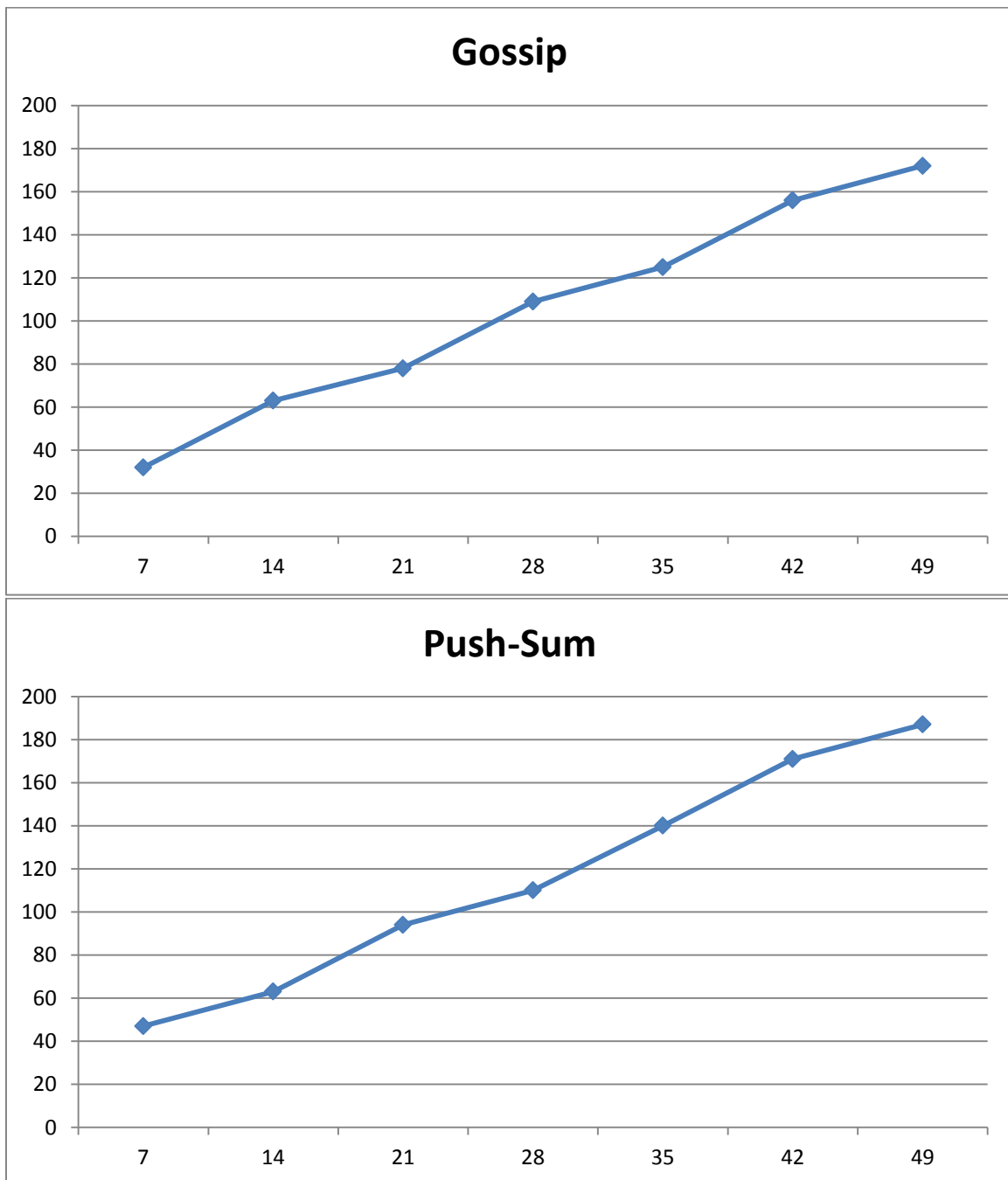
- A similar convergence model was followed for the Push-Sum algorithm.
- If the ratio of s/w for a node did not change more than 10-10 times for 3 consecutive rounds, the node was placed in the list of non-working nodes by the Supervisor Node.
- A non-working or passive node cannot send any messages and participate in the gossip.
- The following algorithm converges if a set number of nodes are placed in the passive list according to the topology selected and the main process is killed.
- Convergence formula similar to that of Gossip.

Implementation Details

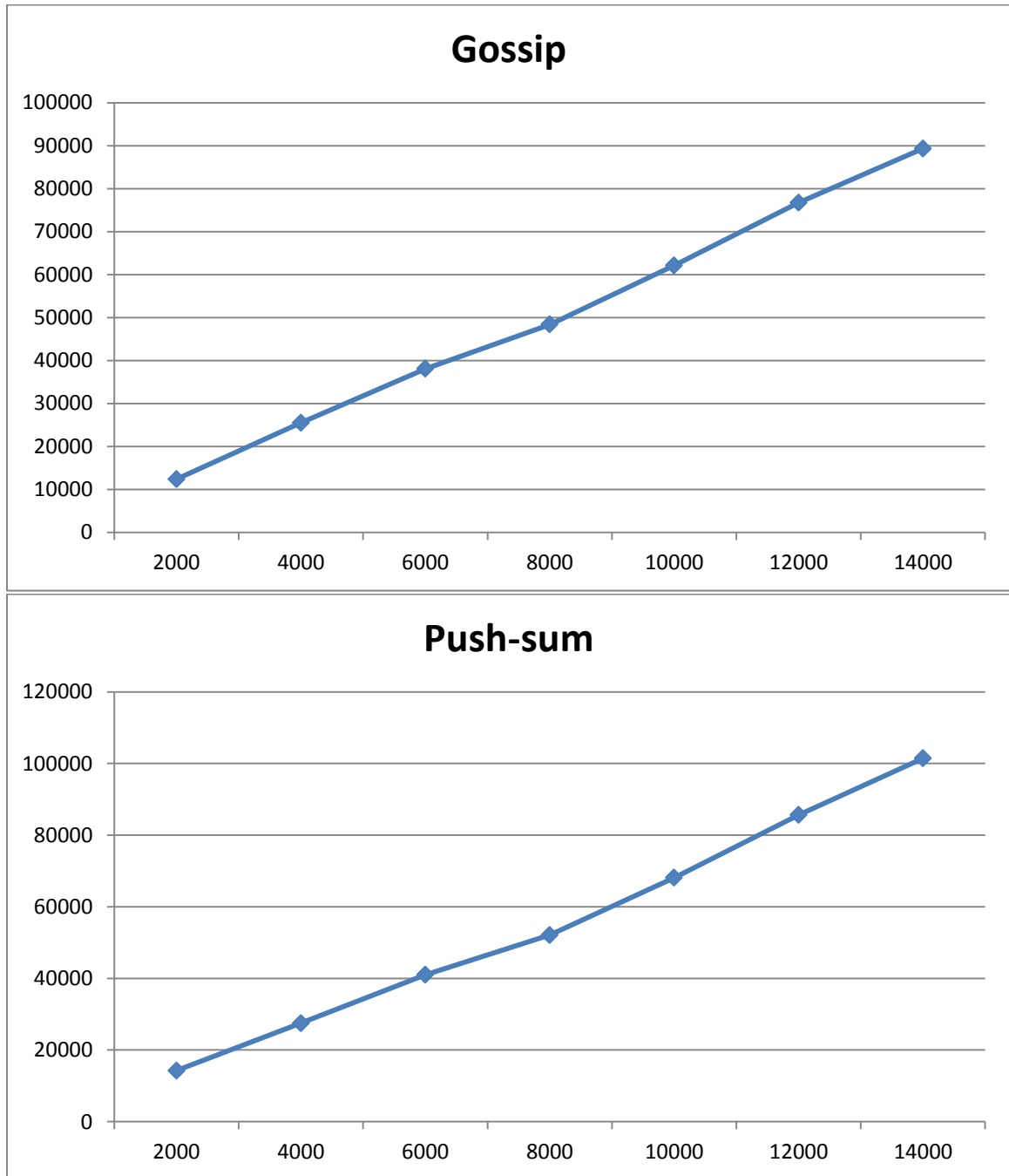
- **project2.ex**: This file gets the number of nodes for the topology and the algorithm from the user.
- **supervisor.ex**: This file creates the list of nodes that have reached the total count and are now not working. These are removed from the topology and now cannot send and receive messages.
- **gossip.ex**: This file implements the gossip algorithm on the given topology.
- **pushsum.ex**: This file implements the pushsum algorithm on the given topology.
- **topology.ex**: This file contains all the six topologies - Line, Imperfect Line, Rand2D, 3D grid, Torus and Full Network.

Graphs (x-axis: Total Actors, y-axis: Time in milliseconds)

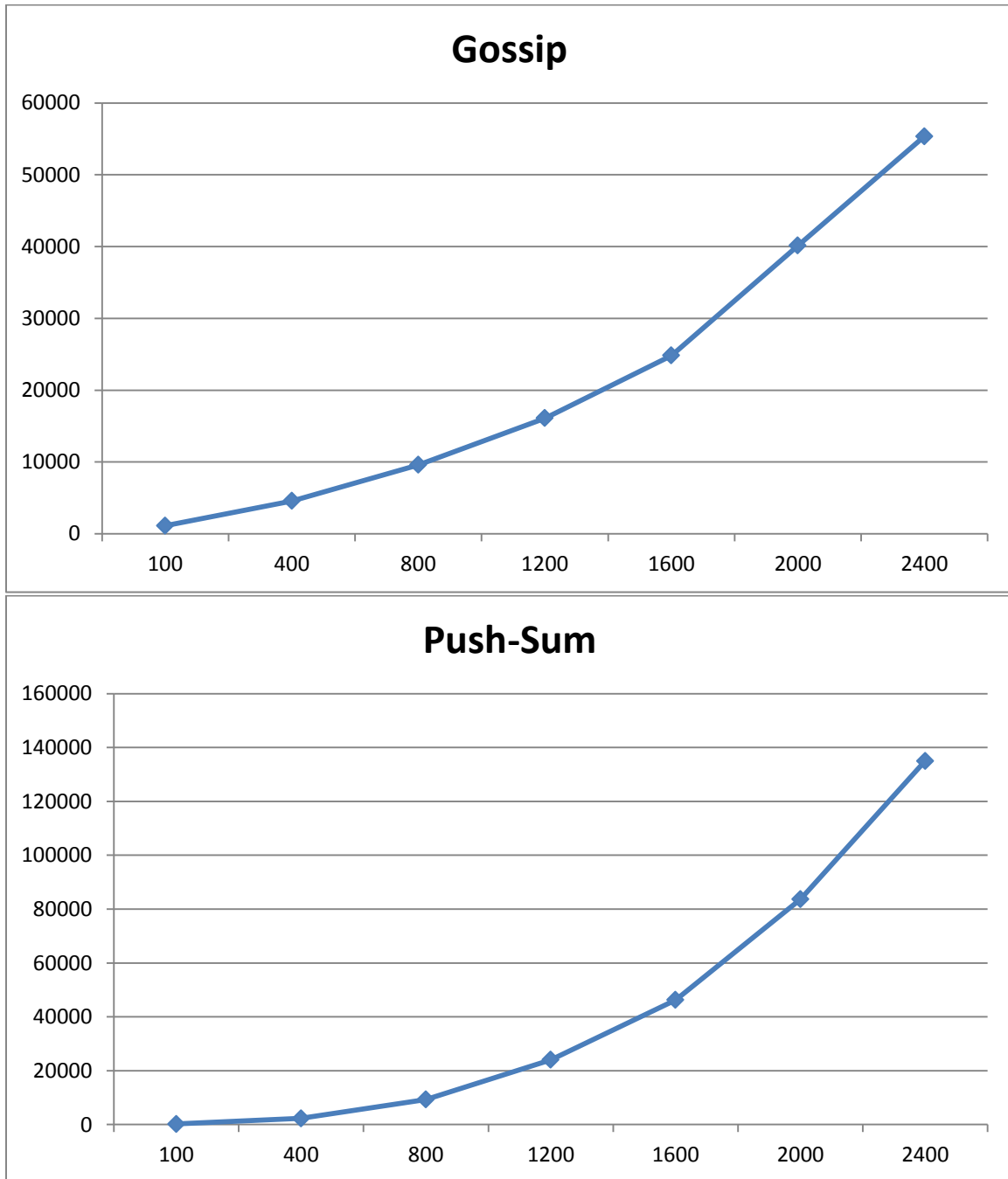
1. Line Topology



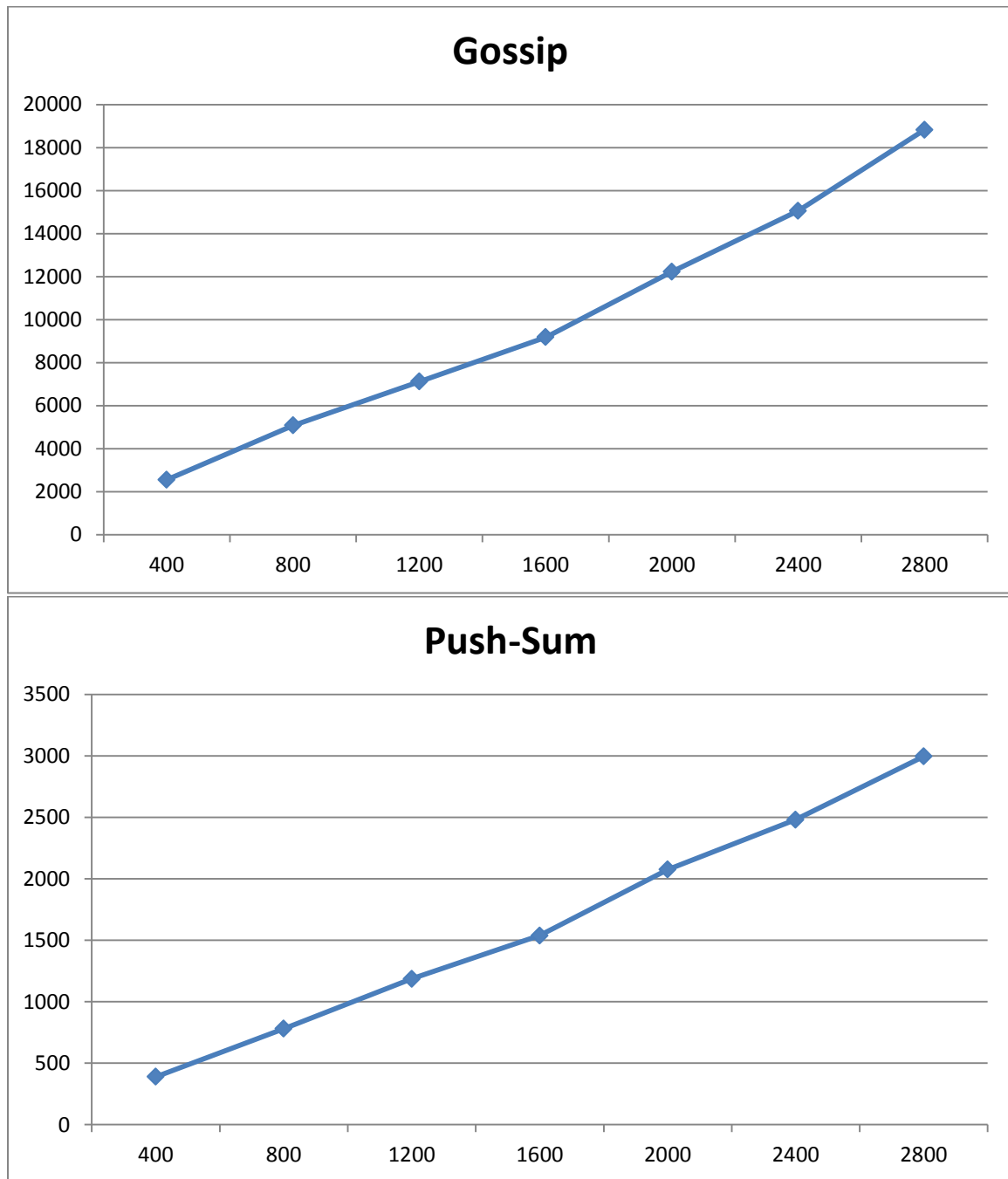
2. Random 2D grid topology



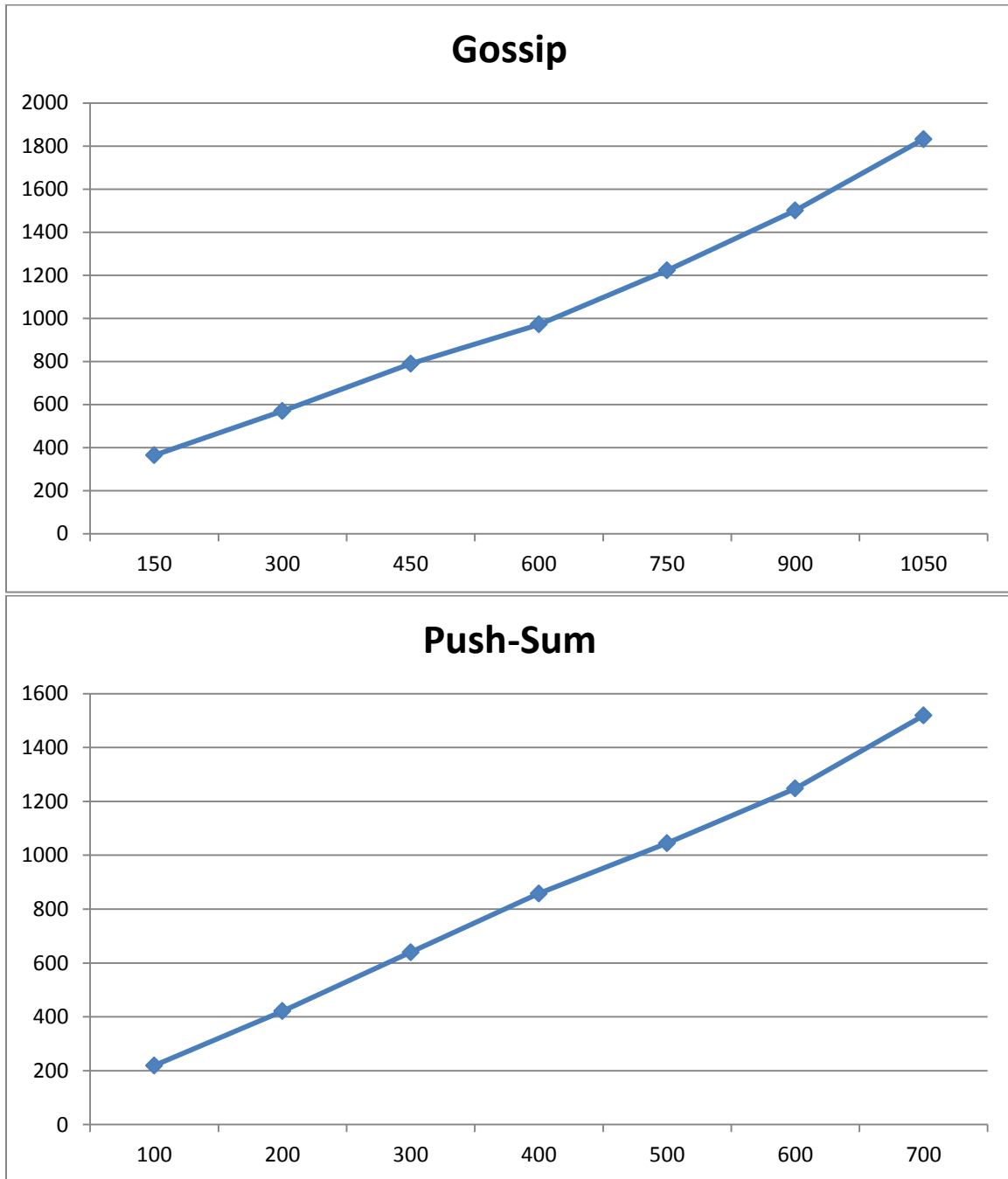
3. Full Network topology



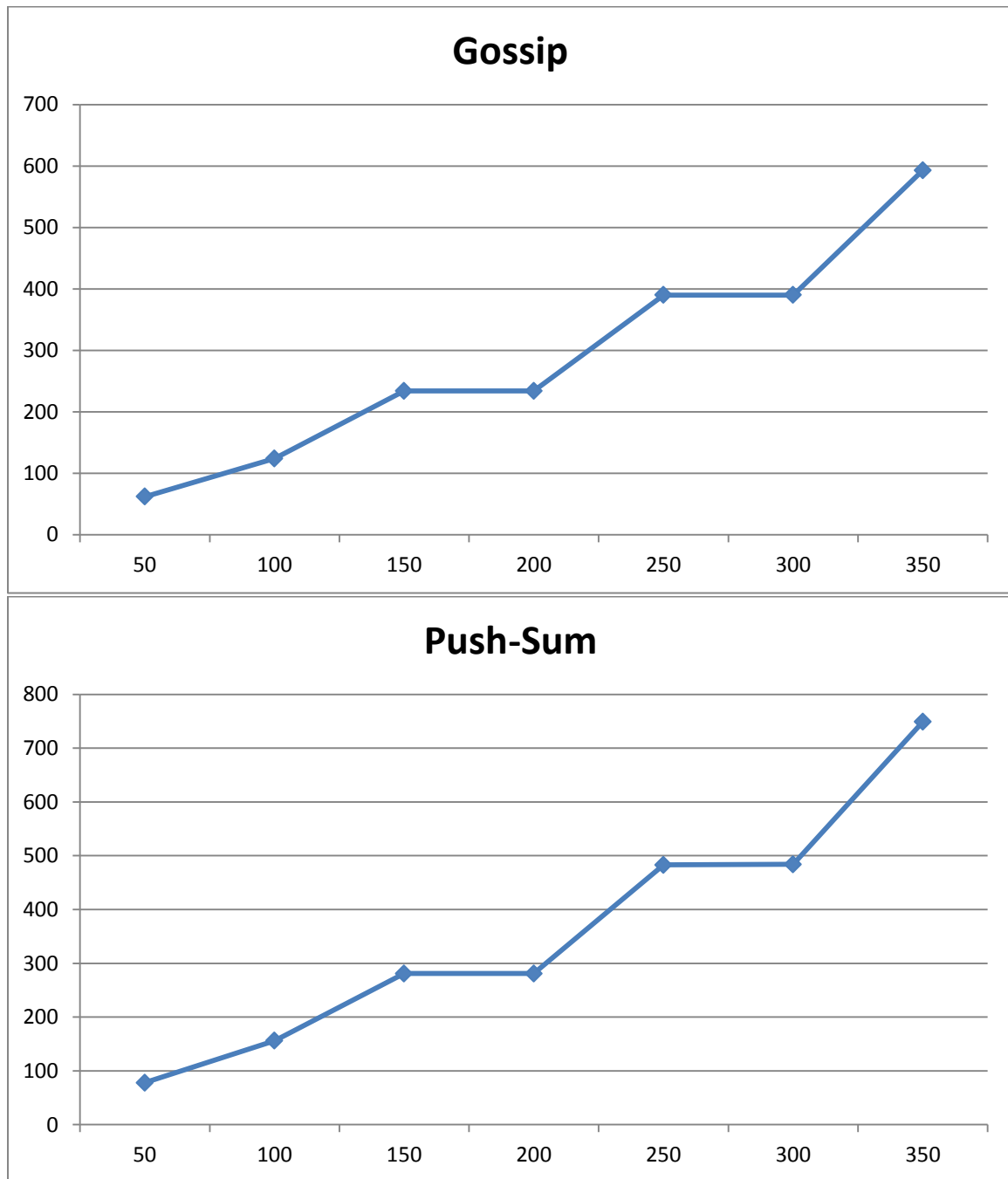
4. Imperfect line topology



5. Torus



6. 3D topology



Interesting findings

- The nearby ranges in the graph of the **3D grid** shows small growth since the sides of the 3D grid is cube root of the number of the nodes given.
- From the graphs, it is seen that **full topology** has the highest probability of converging.
- **3D** and **torus** converges at an average rate since it has max 6 neighbors and 4 neighbors respectively.
- **Line** has the lowest probability of converging. This is because in line, torus and 3D there is a higher chance that the neighbors of a particular node have reached their count and are not participating in the algorithm anymore. In such a case, the algorithm cannot converge. Hence, our ideal topology is: Full topology. Random 2D grid and imperfect line get random neighbor, so they have a better rate of converging.