Distributed Systems Project 2
Gossip Simulator

Gopichand Kommineni (0305-5523)
Hemanth Kumar Malyala (6348-5914)

Steps to run the application:

1. In shell, navigate to the directory where mix.exs exists

2. Run command

>mix escript.build

[Ignore any warnings that you notice]

>./project2 <numNodes> <Topology> <Algorithm>

<numNodes> can be any number

<Topology> can be any of full, line, random 2D, torus 3D, honeycomb

<Algorithm> can be any of push-sum or gossip

Example: > escript my_program 400 pushsum line

After step2, the program will start running based on the topology and algorithm chosen and terminates when the convergence is reached

3. The time taken to achieve convergence is printed on the screen before the program exits.

Application Details:

Our application consists of three files – my_program.ex, Actor1.ex and super1.ex.

- My_program.ex is used to read the command line arguments and implement the escript.
- super1.ex contains the main module which is used for getting the required inputs from the user, build topologies for the created actors, bind neighbor actors to each actors and call Gossip or Push-Sum algorithm.
- Actor.ex contains the actual implementation for creating the actors, bind neighbors with actors, gossip propagation and push-sum calculation.

2. What is working?

Topology Creation:

As soon as the program starts, based on the topology chosen, the main module creates the required number of Actors and binds its neighbor actors with each actor. For ‘full’ topology, every actor is a neighbor of all other actors, that is, every actor can talk directly to any other actor. In ‘line’ topology, all the actors are arranged in parallel lines. Each actor has only 2 neighbors one left and one right, except the first or last actor. For ‘random 2D’ topology, each actor has a pair of (x, y) co-ordinates. All the actors which are at less than 0.1 are considered as its neighbors. In case of “Honeycomb”, each actor has three neighbors except the ones at the corners. A group of 6 actors form a hexagonal structure and they are interconnected to form a honeycomb structure. A “torus 3D” structure is a surface of revolution generated by revolving a circle in three-dimensional space about an axis that is coplanar with the circle. Any actor in this structure is connected to 6 other actors at all instances.

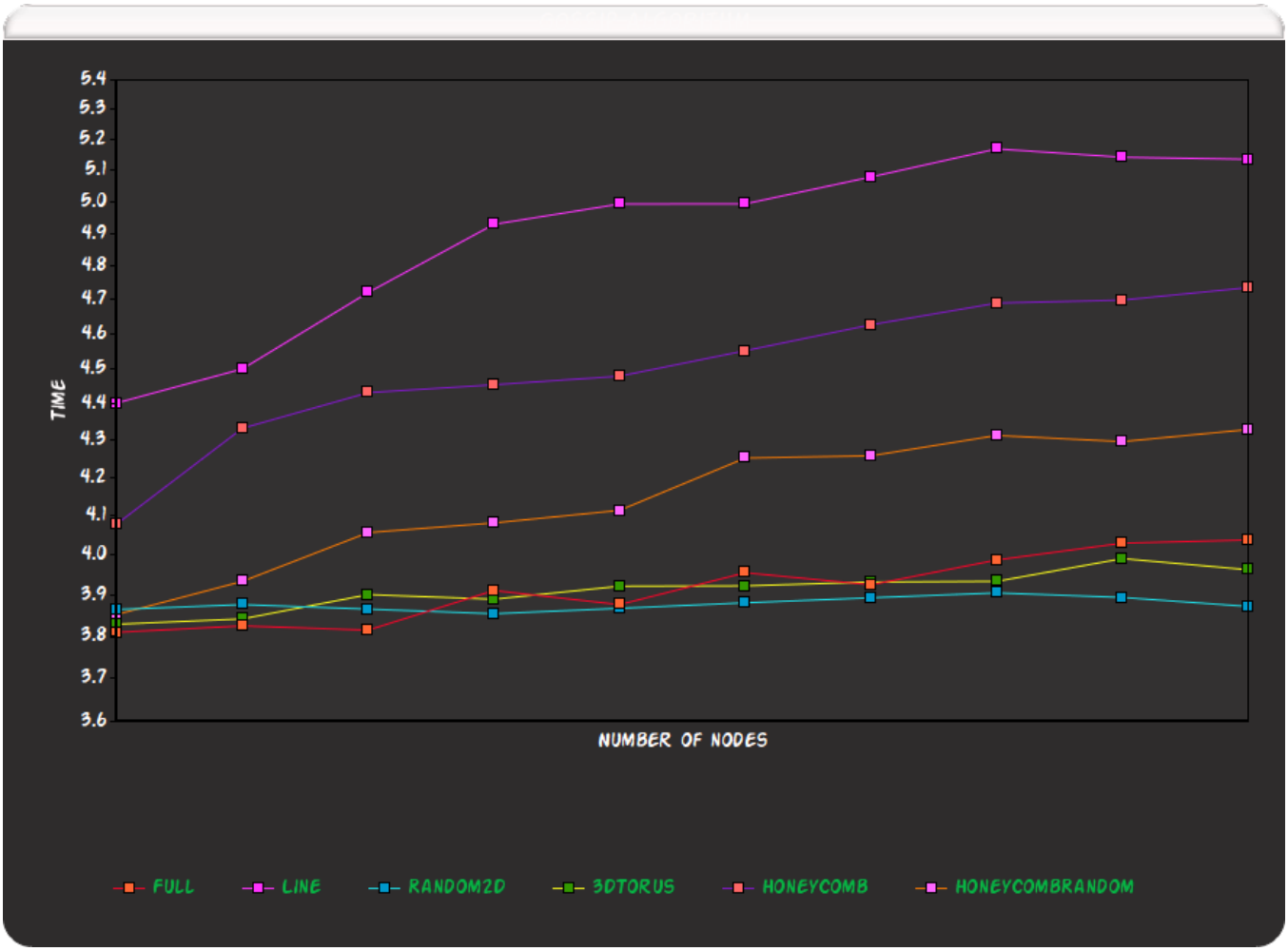
GenServer for managing actors:

We have used Genservers in Actor1.ex to create the Actor processes, maintain states like process ID, s value,w value, neighbor list of a particular actor, and counter for storing the threshold value.

Gossip Algorithm:

Number of Nodes	Topology	Time (in milliseconds)
100	Full	6437
200	Line	31531
300	Random 2D	7328
400	3D Torus	7750
500	Honeycomb	30062

Performance of Gossip Algorithm



Interesting fact:

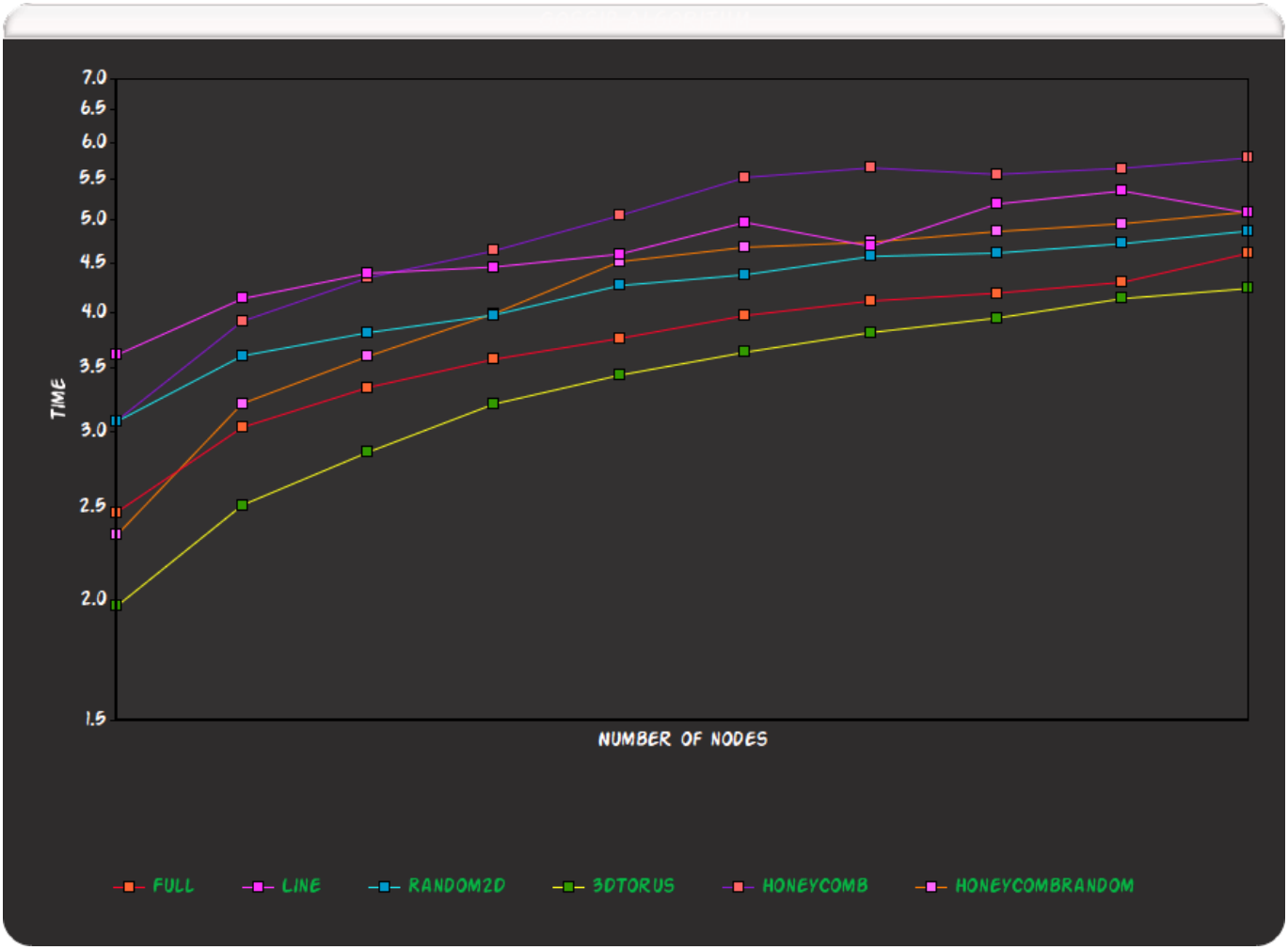
The interesting fact observed here is that the line has at most two neighbors and hence it takes too long to get converge as compared to other topologies.

Also, the performance of topologies is almost independent of the number of the nodes because the gossip algorithm is asynchronous.

PUSH-SUM ALGORITHM:

	Full	Torus3D	Rand2D	Line	Honey Comb	Random HoneyComb
Number of Nodes	1000	1000	1000	1000	1000	1000
Time to Converge (In milliseconds)	40328	17063	72078	118719	616657	121765

Performance of Pushsum Algorithm:



Interesting fact:

The convergence rate for all the topologies in push-sum is higher than that of gossip.

Higher the connectivity of the nodes, better is the performance.