**COP5615 Distributed Operating System**

**Project 2**

**README**

**# Group Members**

1. Bhavya Kavdia, UFID : 7017-7277

2.Ishita Trivedi, UFID : 6893-6496

**Problem Statement -**

As described in class Gossip type algorithms can be used both for group communication and for aggregate computation. The goal of this project is to determine the convergence of such algorithms through a simulator based on actors written in F#. Since actors in F# are fully asynchronous, the particular type of Gossip implemented is the so called Asynchronous Gossip.

**Maximum no of nodes** considered for Gossip algorithm are **100000** and for Push- Sum algorithm is **50000**.

Note – This file contains command to run project 2.

**Input –**

1. numNodes – No of nodes in the network
2. topology – Can be one of the networks – full, line, 3D, imp3D
3. algorithm – Can be one of the two – gossip or push-sum

**Command to run –**

1. To run Model 1 :

dotnet fsi –langversion:preview Project2Model1.fsx <numNodes> <topology> <algorithm>

1. To run Model 2 :

dotnet fsi –langversion:preview Project2Model2.fsx <numNodes> <topology> <algorithm>

**Implementation –**

* **Steps :**

1. We built the network topology by constructing neighbor list for each actor during its creation. The neighbor list consists of only the remaining nodes present in the network.
2. Then we begin the algorithm by randomly selecting a single actor or participant.

* **Algorithm implementation :**

1. The information for the neighbors of an actor are stored in a list which is created during the initiation.
2. Once a rumor message is received from one of the neighbors in the neighbor list, the actor then sends a message to one of its own neighbors from its neighbor list and also maintains its own state i.e. a message count in which it tracks the no of times the message is received by incrementing the count by 1.
3. Also, an Actor Dispatcher periodically sends rumor message to itself from the first time it receives message till the time it terminates.

* **Termination:**

The program terminates when all the actors receive the message the number of times we have set the exhaustion limit to be.

* **Output:**

The amount of time it takes for the program to converge in milliseconds with any given no of nodes for any given network topology.