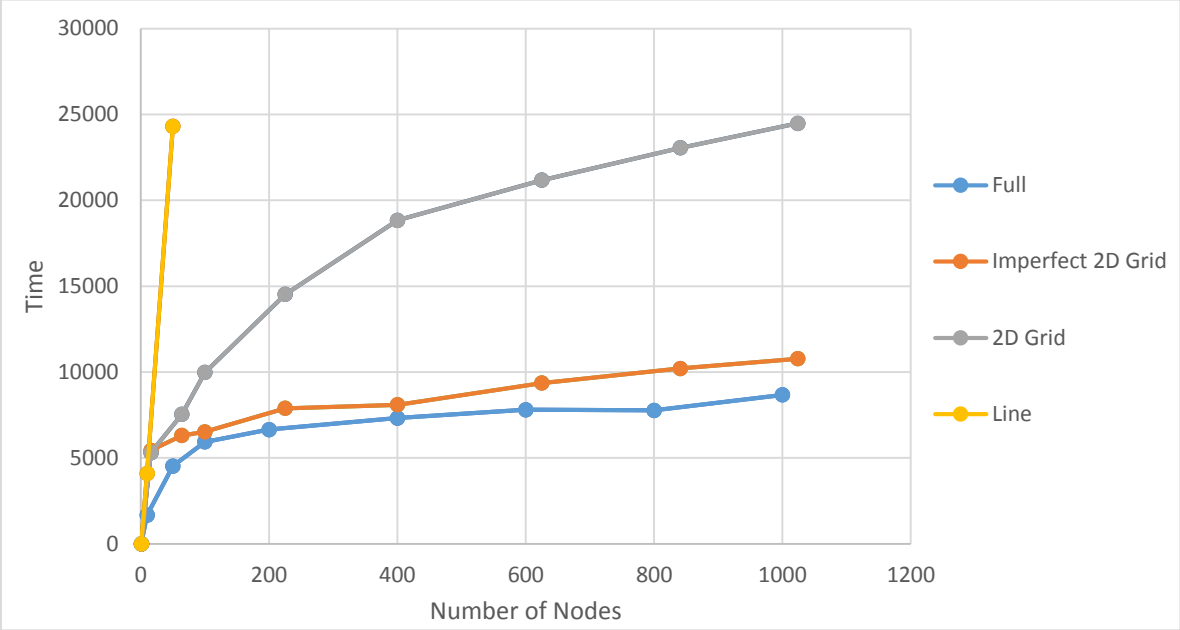
COP 5615 Project 2 Report

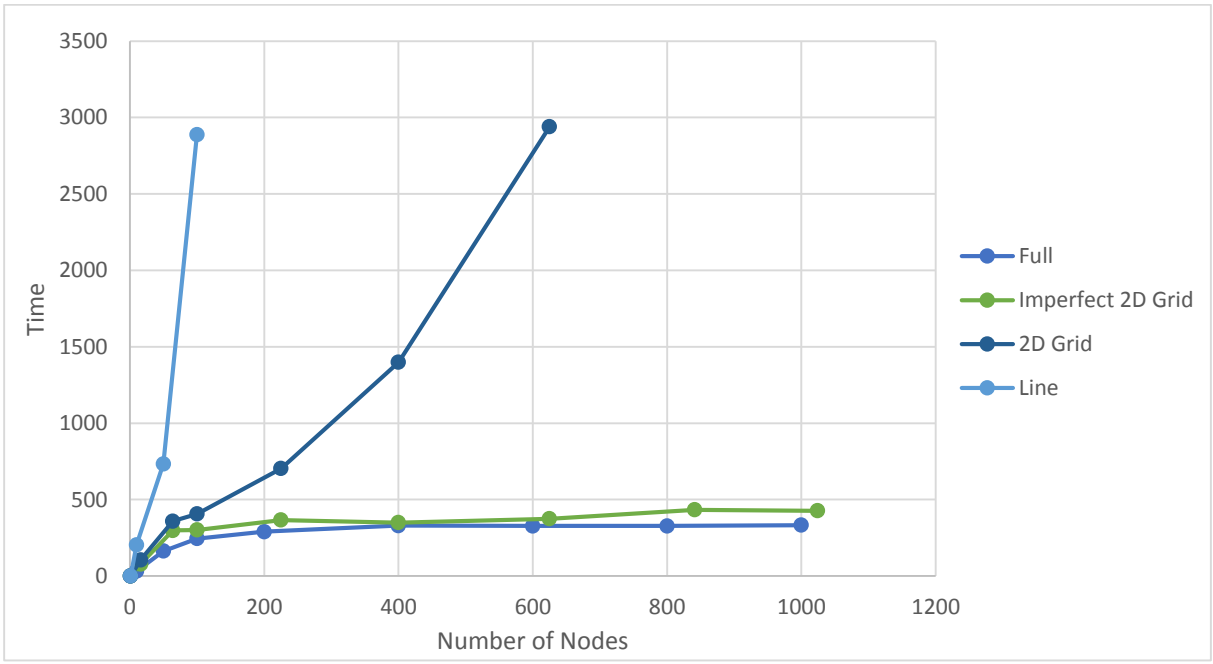
Yang OU(UFID: 2012-6327)

1. Dependency of convergence time as a function of size of network with Gossip Algorithm



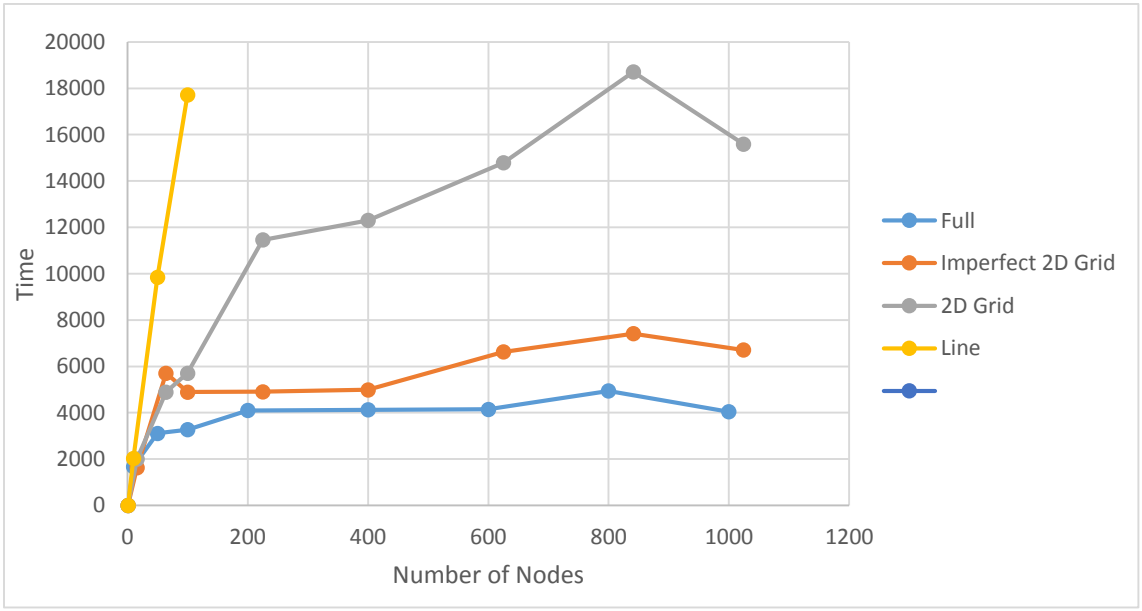
From the above plot, we can see that network with full topology gives the shortest convergence time and is roughly in O(log(N)). And then follows imp2D grid, 2D grid and line. And line topology has the longest convergence time and is roughly in O(n2) time.

2. Dependency of convergence time as a function of size of network with Push-Sum Algorithm



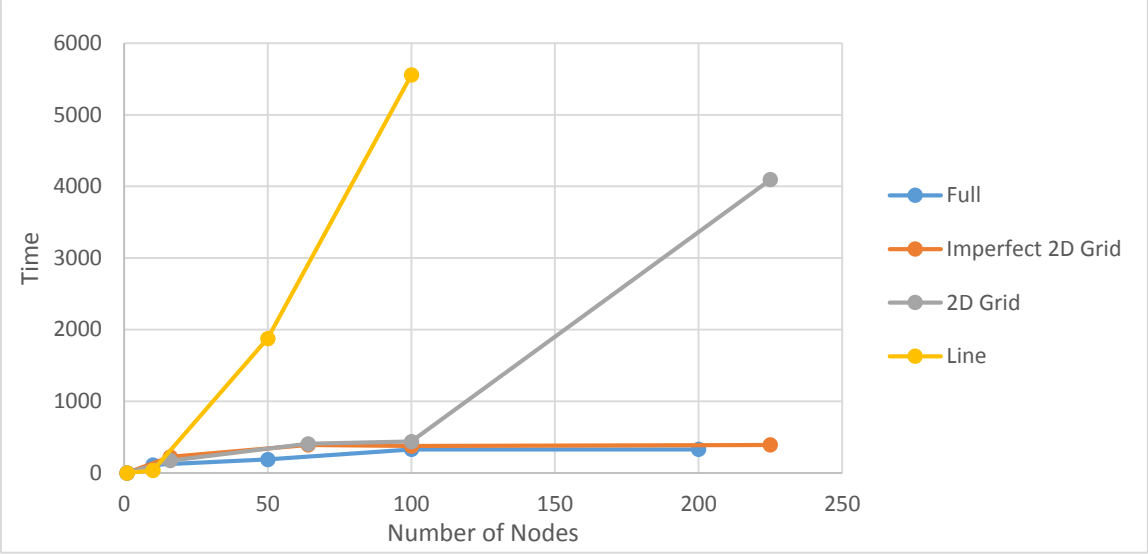
From the above plot, networks with full or imperfect 2D topology are very fast. Then comes 2D and line. And the converged value is fixed. And I implemented another version of project2more.scala in folder peoject2more. In this version, I tried to achieve convergence of more nodes with a round-send-mechanism. And the output will show the information about ratio of converged nodes. In this version, the time will not follow strictly the above results because in many case some nodes just stop if their neighbors are all converged. The command to run push-sum is the same. Here I use "bad-converge" for the nodes stopped not due to s/w value converge but due to no neighbor. And I observed the convergence ratio (good-converged nodes/total number of nodes) goes down from full, imp2D, 2D to line as 100%-84%-70%-59%.

3. Failure Mode
   (1) In failure mode, the system will have ONE dead node for EVERY specified period (200 milliseconds as default). And a new actor will be created to take its place which will be explained below. The output will have the time and the number of failed nodes. Some execution in failure will simply stuck. And most executions can have a result. And the following plots are drawn based on successful executions.
   (2) To use failure mode: Compile "project2bonus.scala". Execute using command "scala project2bonus numNodes Topology Algorithm Failure-Gap"
   e.g. "scala project2 100 full gossip 200" This example means ONE node will die EVERY 200 milliseconds.
   (3) In failure mode, if one node dies, the master will know and will trigger the following operation:
   Check whether the actor has already converged. If so, simply do nothing.
   If the node hasn't converged, the neighbor of dead actor will remove it from sending list. Then master will create a new actor to take its job. The new actor will have the same information as before. Master then tells the network of the new born actor by adding the new actor's address to the dead actor's neighbor.
   (4) For GOSSIP algorithm with ONE node dying EVERY 200 Milliseconds. The plot is as follows. The failure gap can be changed using the command mentioned in (1).



We can see the convergence times of gossip algorithm are roughly the same.

(5) For Push-Sum algorithm with ONE node dying EVERY 200 Milliseconds. The plot is as follows. The failure gap can be changed using the command mentioned in (1).



For Push-Sum, things get really complicated. Because the new node created for dead node can't maintain the old s/w value. So the whole system is facing the convergence problem. And if number of nodes becomes larger, time will grow, as well as dead nodes. For larger network, it is really HARD for the system to converge.