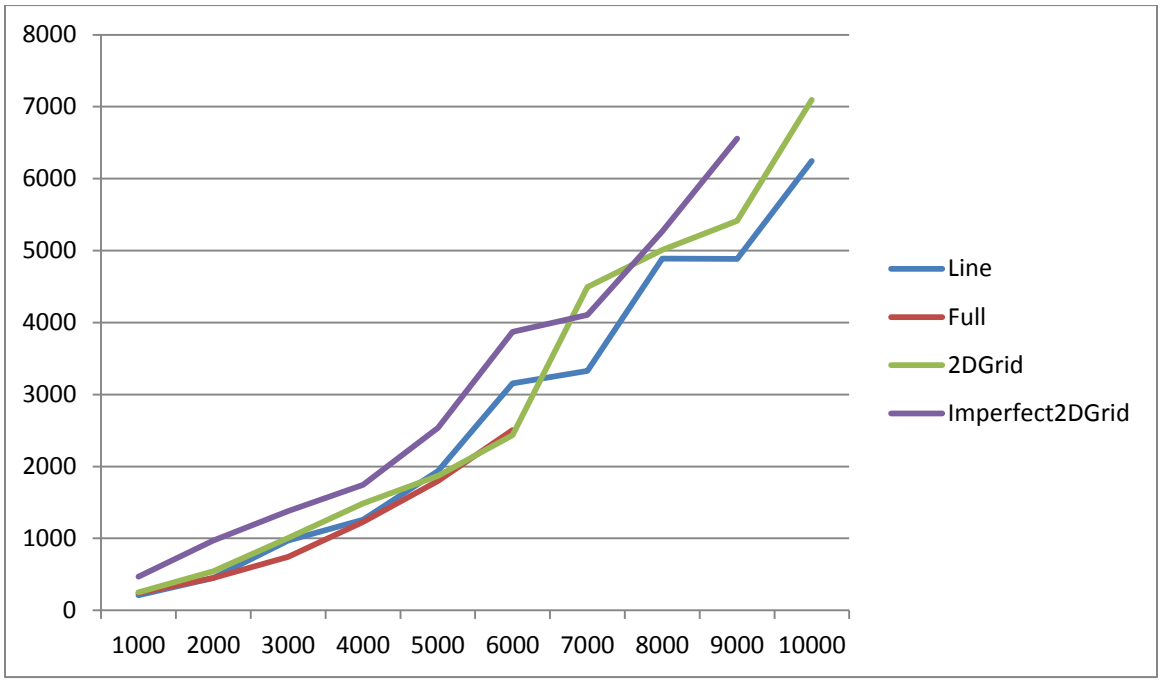
REPORT

GOSSIP ALGORITHM FOR INFORMATION PROPAGATION

Below is the graph plotted against
X-axis : Number of Nodes
Y-axis : Convergence Time
For the algorithm **GOSSIP**



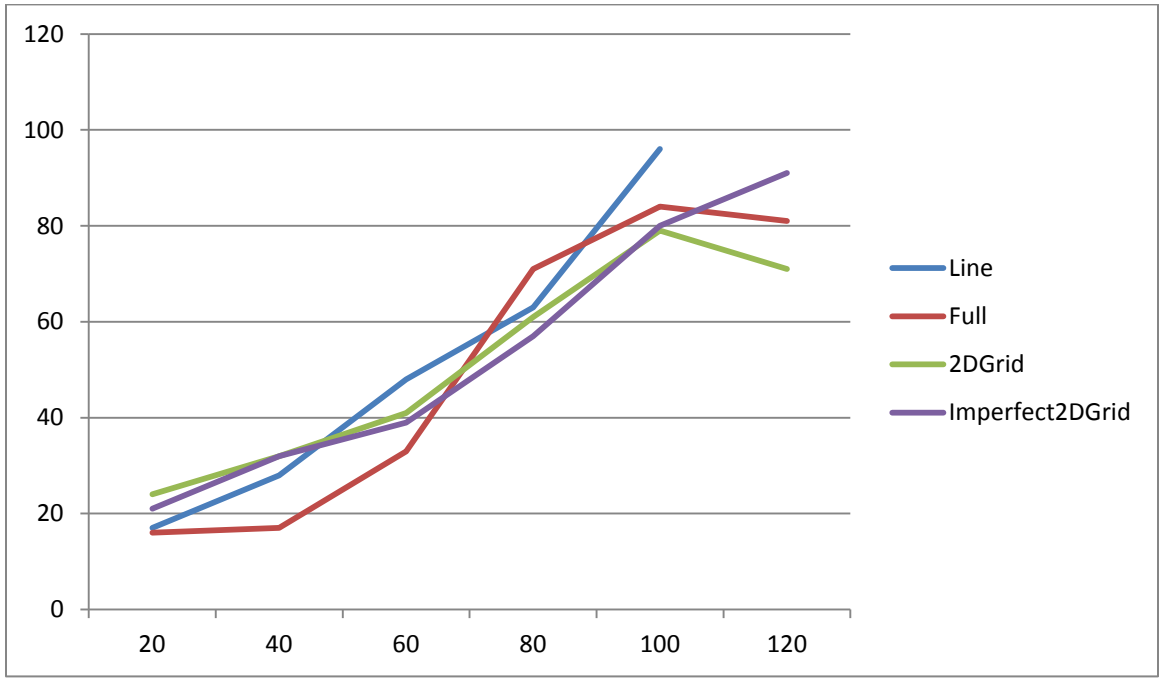
Gossip protocol runs on the concept “random peer selection with a given frequency”, where each machine picks other machine in random and shares the rumor. Gossip protocols are the most efficient ones often used in modern distributed systems which makes communication faster and easier for extremely large underlying networks.

Above graph explains the performance of the Gossip protocol for four different topologies Line, Full, 2DGrid and Imperfect2DGrid. For each topology using gossip algorithm the message is being converged with some limit on the number of nodes. As the program uses a threshold value of number of times each node can listen the rumor that particular node stops transmitting sending the rumors further. This threshold value leads to the situation where a deadlock may occur while transmitting the messages. Thus the upper boundary condition on the number of nodes parameter is a valid case. In Line topology as each node consists of only two neighbors the probability for a deadlock to occur is high, which decreases the possibility to converge the message. The convergence time increases with increase in number of nodes for all topologies. For Full topology every other node of the network will be its neighbor, so the probability of selecting a random node is higher in this case, even when some nodes has already reached their threshold value. Hence a message converges easily in such a type of topology.

Similar in case of other two topologies which are 2DGrid and Imperfect2DGrid, they have neighbors as nodes present in the nodes grid and an extra random node for Imperfect2DGrid. There may exists cases where a node never receives any rumor at all from its neighbors for the entire phase, this is the reason for the definition of convergence to be a situation where maximum number of nodes (say 90%) receives the rumor.

PUSHSUM ALGORITHM FOR SUM COMPUTATION

Below is the graph plotted against
X-axis : Number of Nodes
Y-axis : Convergence Time
For the algorithm **PUSHSUM**



Push Sum algorithm uses the Gossip scheme of communication for the computation of aggregates. With the Push-Sum algorithm, each actor starts with two values – s-value and w-value. The s-value for actor i is the value i and w value for each actor is 1. The Push-Sum algorithm first selects a random node and sends it the PushSum message. On receipt of the PushSum message, the actor computes the new values for s and w by adding it's previous values with the new values. Half of the newly computed value of s as well as half of the newly computed value of w are transmitted by this actor to its random neighbors. The remaining haves are kept with the actor itself. The ratio of s to w value is computed at each step and if for a particular actor this ratio is not found to change by more than 10^(-10) in 3 consecutive rounds then this actor is made to terminate. The first time an actor receives a message, its id is noted down in a list and checked if the number of actors receiving the message at least once is greater than 90% of the total number of actors. If it is true, the algorithm is said to converge and the system shuts down. As the number of nodes increases, the time to converge for the different topologies also increases as can be seen from the graph. For very high number of nodes, the system can sometime enter into a deadlock

situation which has been recorded by the maximum number of nodes dealt with for each type of topology.