Tapestry Algorithm(Project-2 Bonus)

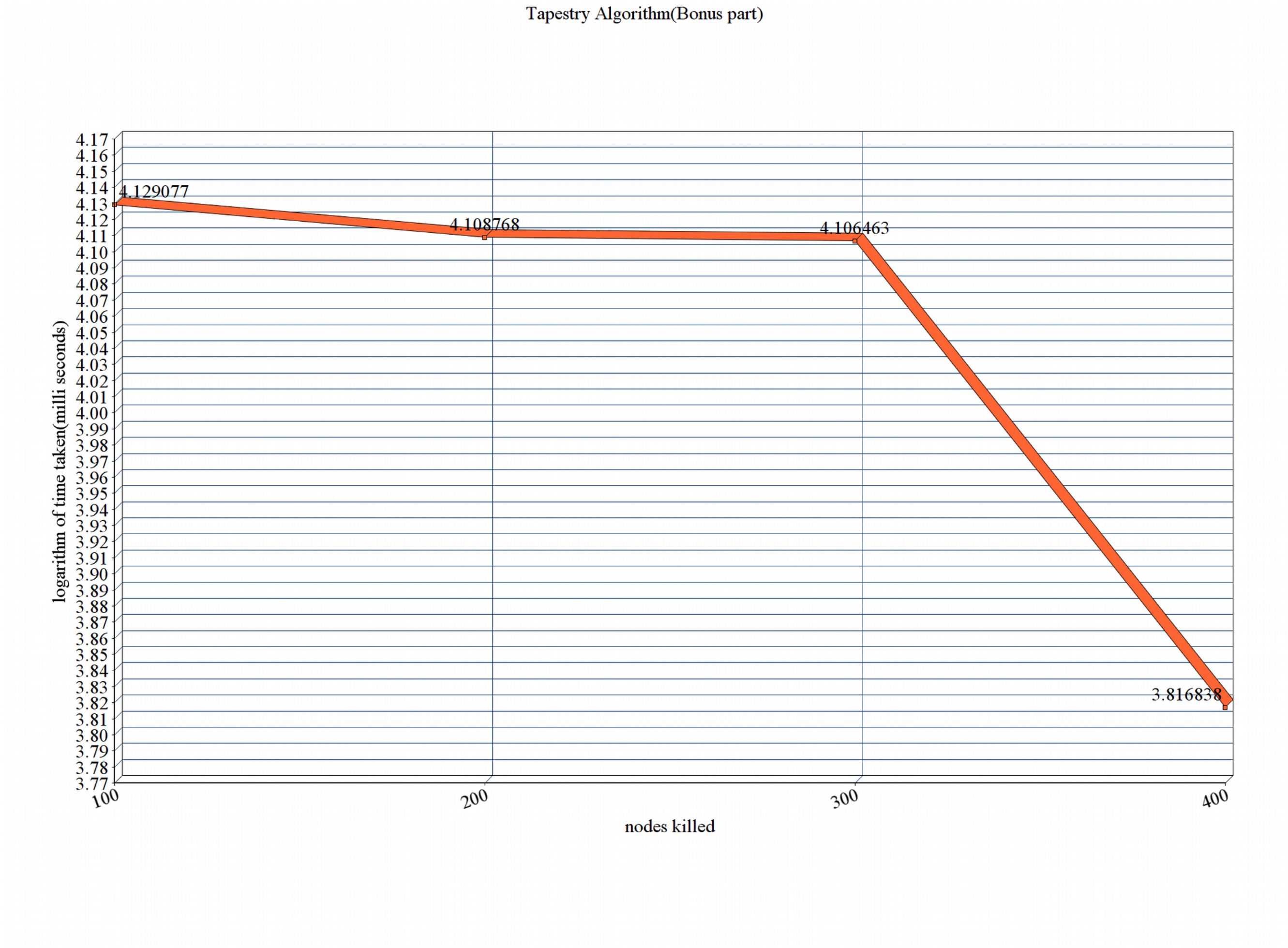
Distributed Operating Systems

Group Members:

Chandan Chowdary (UFID-6972-9002)
Gayathri Manogna Isireddy (UFID-9124-0699)

Project Description and Approach:

To implement this algorithm we have created some nodes in the network. Routing tables of all the nodes in the network are created along with the dynamic nodes just like we calculated in part 1(non bonus part). To make the system fault tolerant, there is a slight modification in the approach. In the earlier approach, a single matching node in inserted at the suitable position in the routing table of each node. But in this case a list of nodes is inserted at each position in routing table. If a node that is killed is picked from this list as a next hop, then we are checking if the node is killed or alive. If the node is alive we pick up the node and proceed with the algorithm, else the node will be dropped and another node will be picked up from the list. In this way whenever a killed node is picked as next hop, that node is dropped and another node which matches with the destination node from the same list is picked up and sent as a next hop. If a destination node is killed, then the hops that are calculated till the level before it reached destination is stored and is returned.



The graph that is shown above is drawn between the logarithm of time taken to perform all the requests after killing the nodes which are shown on x axis.

Observations:

The time taken is dependent on the number of nodes that are left in the network after killing the nodes.
The system achieves the convergence irrespective of number of nodes that are killed.
Even if the destination node is killed the system will continue to work as it would return the hops that it took to reach the level before it reaches the destination.