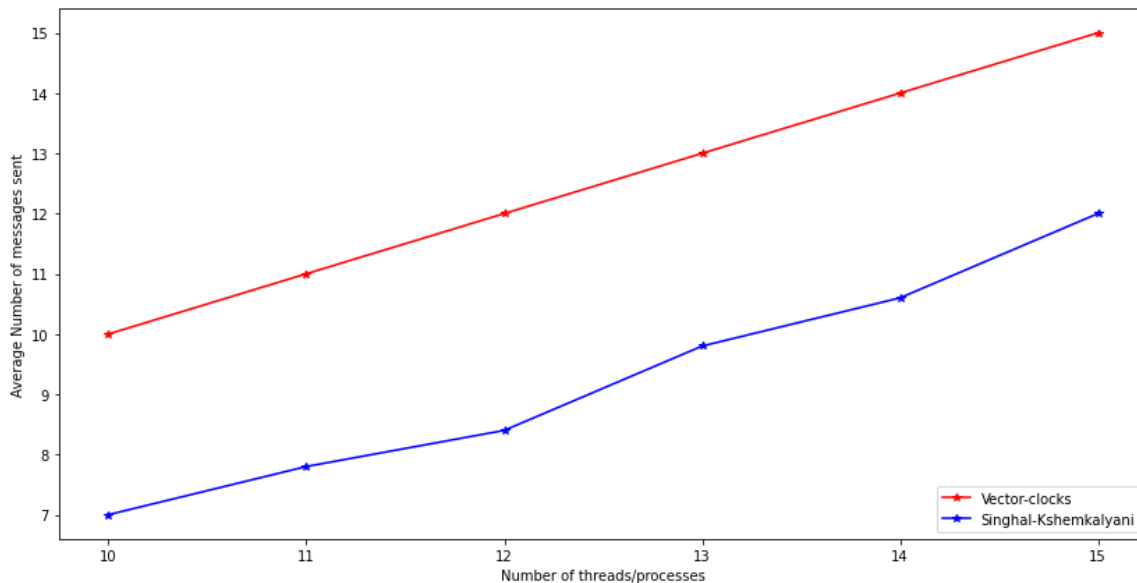


Report

The topology I have considered is a mesh topology that every thread/process can send message to all other processes/threads. In both the programs (Vector clock and Singhal-Kshemakalyani) I have sent the vector/vector pair by converting it to a string and sent it using character buffer.

Graph plot between the average number of entries send between different number of threads/processes is attached below.



Note: The pair sent in each message in case of Singhal-Kshemakalyani model is considered as a one entry.

Analysis of graph: I basically run the vector clock program and Singhal-Kshemakalyani program **5 times** and calculated the average number of messages sent when the number of threads varied between 10-15. If we see that in vector clocks, the average number of messages sent are fixed and equal to the number of threads but in case of Singhal-Kshemakalyani we have to actually send a smaller number of entries as compared to vector clock and the difference between the number of messages to be sent in both the cases will keep increasing as the number of threads/processes increases.

Vector-Clock

In case of Vector-Clock implementation we are only using a single vector array for each process/thread.

Singhal-Kshemakalyani

In case of this implementation, we are using two additional vectors one is “last sent” and another one is “last-update”.