Report

Author: Mrinal Aich (CS16MTECH11009)

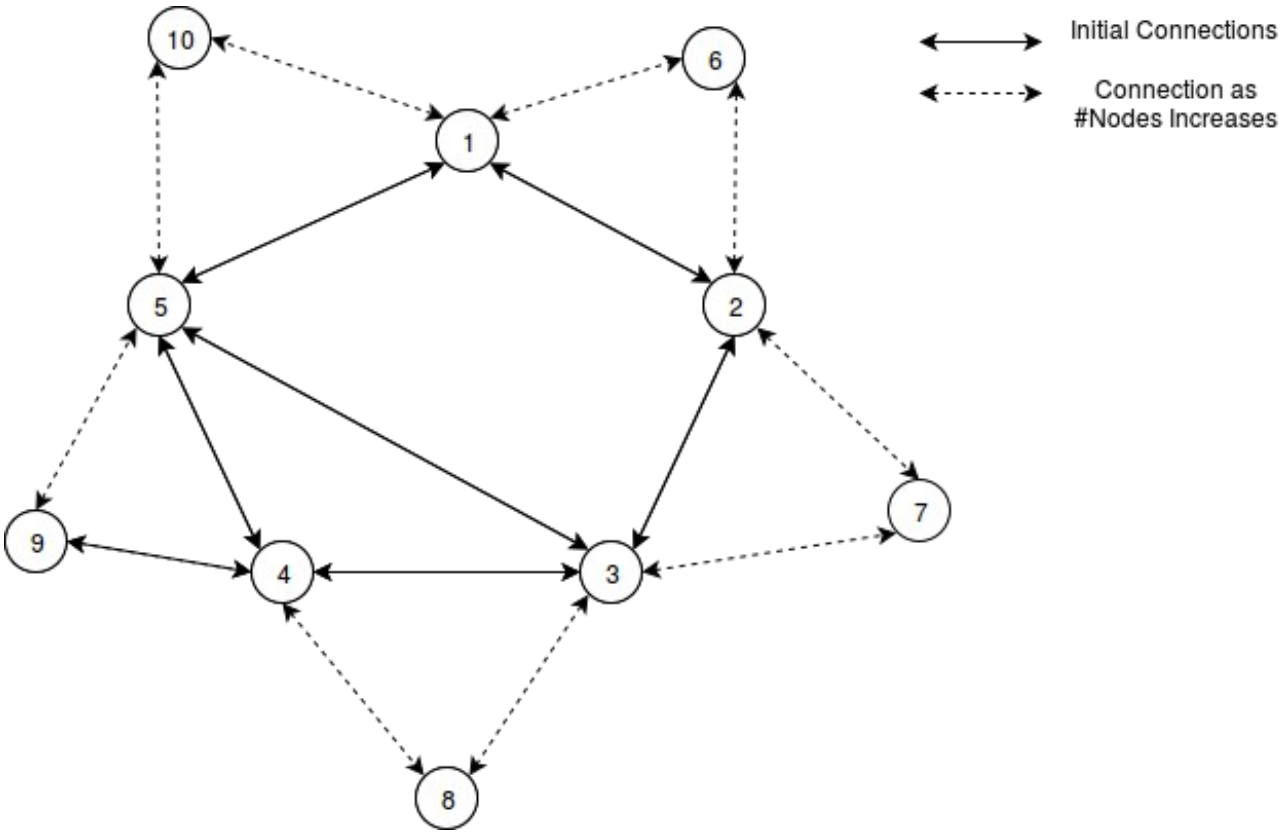
System Model:
The entire distributed network topology is created by considering each Node as a thread. These node-threads further create `server thread` and `client threads` to communicate with their neighbours. Each client thread is connected to a node-neighbour.

Reason for multiple-Client Threads over single-Client Thread:
Single Client Thread : If a message is delayed over an interface, it will block other messages to sent over the other interfaces causing possible bottleneck. Hence, multiple-client thread system model is choosen.

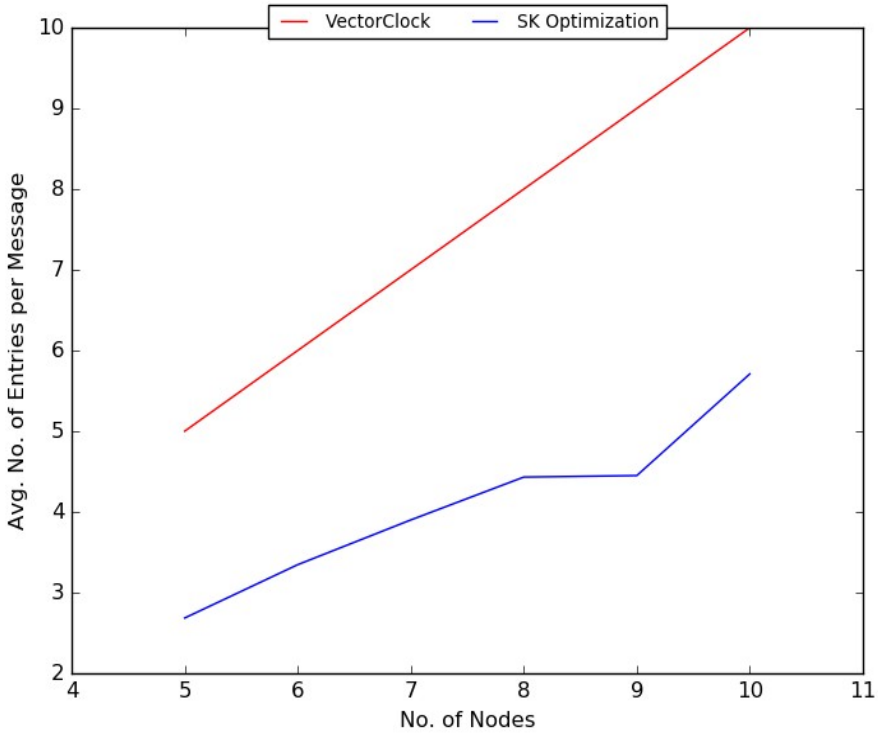
Message Passing:
Node-thread shares a message queue with each of its client-threads.
The node-thread pushes a control message into the queue of its client-thread which is to be sent to its neighbour. The client thread checks for a message and sends it over the TCP connection.

Performance comparision of Vector Clock and SK-Optimization algorithms:

Input Graph Topology:



The graph is plotted by running the code over the two algorithms with different number of nodes across multiple runs.



Storage Space:
Vector clock algorithm – Each nodes maintains a vector clock of the size of the number of nodes in the topology. Hence, storage space is O(n).

SK optimization technique – In addition to the vector clock, each node maintains two vectors `LastUpdate` and `LastSent` of size equal to the number of entries in the topology which are used for sending specific entries of the vector clock. So, the storage space is also O(n).

Analysis:
In Vector clock algorithm, the entire vector clock of the sender process is sent. So, the number of entries will be the same and it increases linearly with the number of nodes in the topology.

In SK optimization technique, between successive messages sent to the same process only a few entries of the vector clock at the sender process are likely to change. Hence, messages contain only those entries of a vector clock that differ since the last message sent to the same process.