**Design document for Graph Algorithm problem:**

**Solution design:**

This document is regarding the design of the Graph algorithm and implementation

The design is done considering each node/process as network devices who can communicate in public or private network.

There are ‘N’ different nodes each representing a process.  These nodes (processes) are running distributed in systems that are in the public network or any network.

Each node or process will read neighbor information and cost from a file thus will generate a minimum spanning tree. Each node will have information about the shortest path from itself (source) to other nodes in the network after reading the input from the file.

**Data structure to store the Alive key**:

This has info about state (UP/DOWN) of each node in the graph.:

long uint32\_t ALIVE\_KEY;

This bitmap is used to store max of 64 nodes. This packet will contain information regarding each node i.e state is  UP/DOWN  and this will be shared/updated with other nodes via heartbeat contain the Alive Key shared between nodes as per the graph.

**Scenarios:**

1. Nodes getting added (started up) to the network, If it has a valid path. This includes addition of new nodes to the existing network topology.
2. It will send the alive packet to the neighbouring nodes. This alive packet will contain the bitmap with its own state and peer connected nodes update signifying that the node is up. This packet is pushed to neighbouring node and then the neighbouring node will update bitmap in the packet  and will forward to its neighbour. In this process  all nodes will be aware of its neighbouring node state(UP/DOWN).
3. Heartbeat messages shared between neighbours

There is a heartbeat mechanism to determine the “UP/DOWN” state of each node. Each node will send heartbeat messages and expects acknowledgement as heartbeat response. A heartbeat packet is sent to its neighboring nodes in time interval of 4 seconds.

A node receives a new masked bit in ALIVE\_KEY in heartbeat packet which will notify that new node has been added. So, we will update node bit in local ALIVE\_KEY and we send the state to peer node . The graph is recalculated each time a state change happens in ALIVE\_KEY.

1. Nodes getting lost from the network . (Process getting killed or not able to communicate with other nodes)

If a node does not receive any heartbeat response from the neighbour node, it will retry for 4 time with incremental time duration of 4, 8, 16, 32 seconds. If all these retries failed than the node state of the neighbor will be masked as down in local ALIVE\_KEY. This updated ALIVE\_KEY will be forwarded to peer node and recalculate the graph.



