Tejas Networks Hackathon

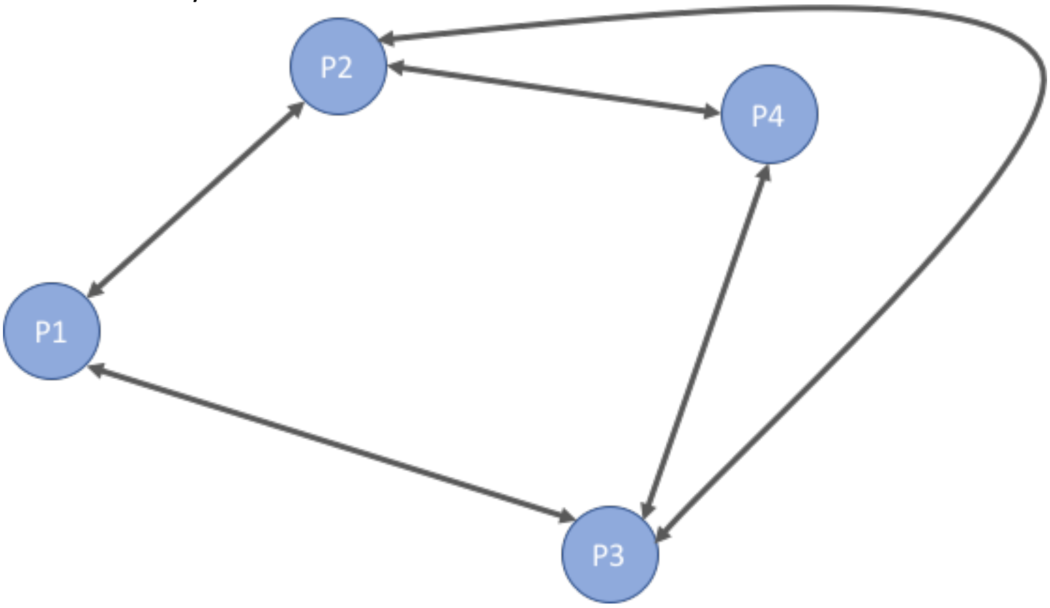
17 June 2017

Participants:
Abir Chaudhuri (abirchaudhuri@gmail.com, +919986827272)
Pradeep Kumar (pradeepkodanya@gmail.com, +919916749494)

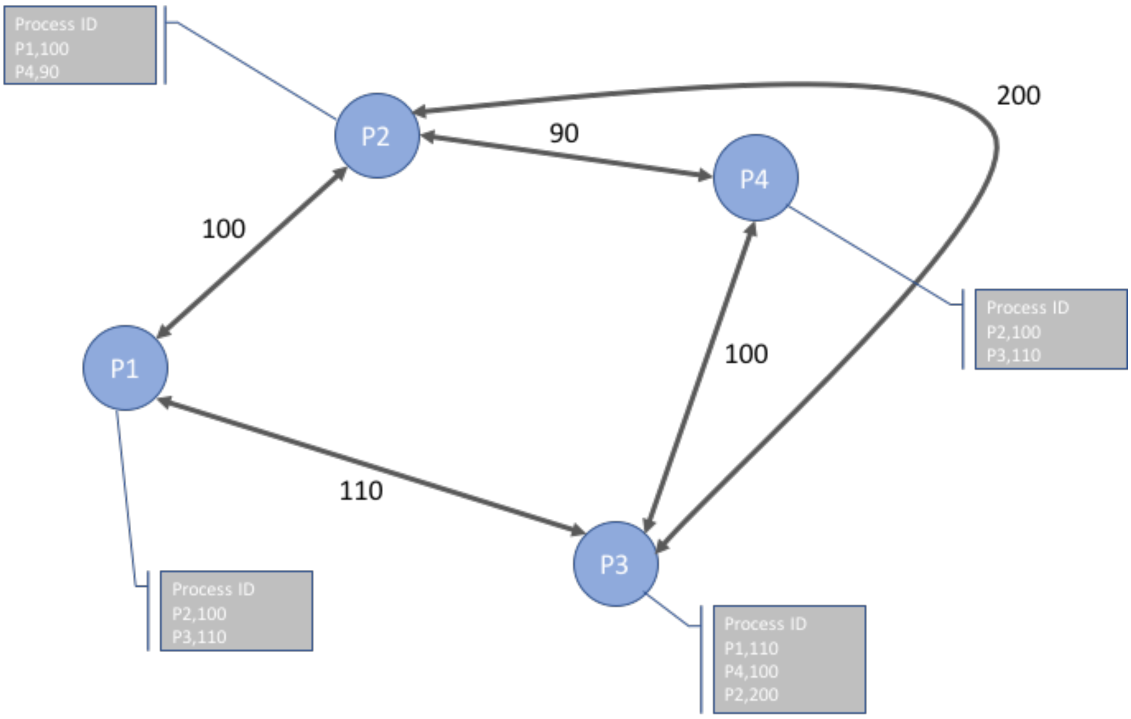
Team name:
Idle Heights

Introduction

The environment consists of 4 processes running as networking nodes. These nodes are connected in the way as shown below.

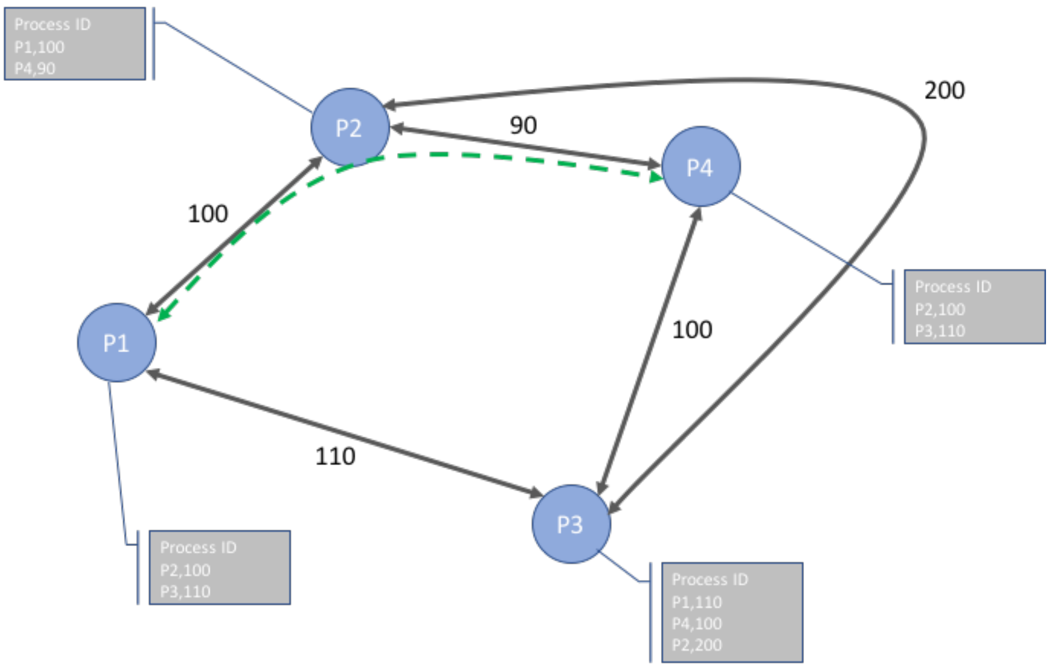


Every node is running independently and can communicate with any of the other nodes. It may have unique path or multiple redundant paths. In case of unique path, communication becomes straightforward but in case of multiple redundant paths, node should be able to decide which path to choose. Every node has a file that is used to maintain information about its neighbors and the cost of edges that are used to connect these neighbors. Cost is used here to understand which path is providing better performance. A path with lower cost will be treated as better compared to a path with higher cost.

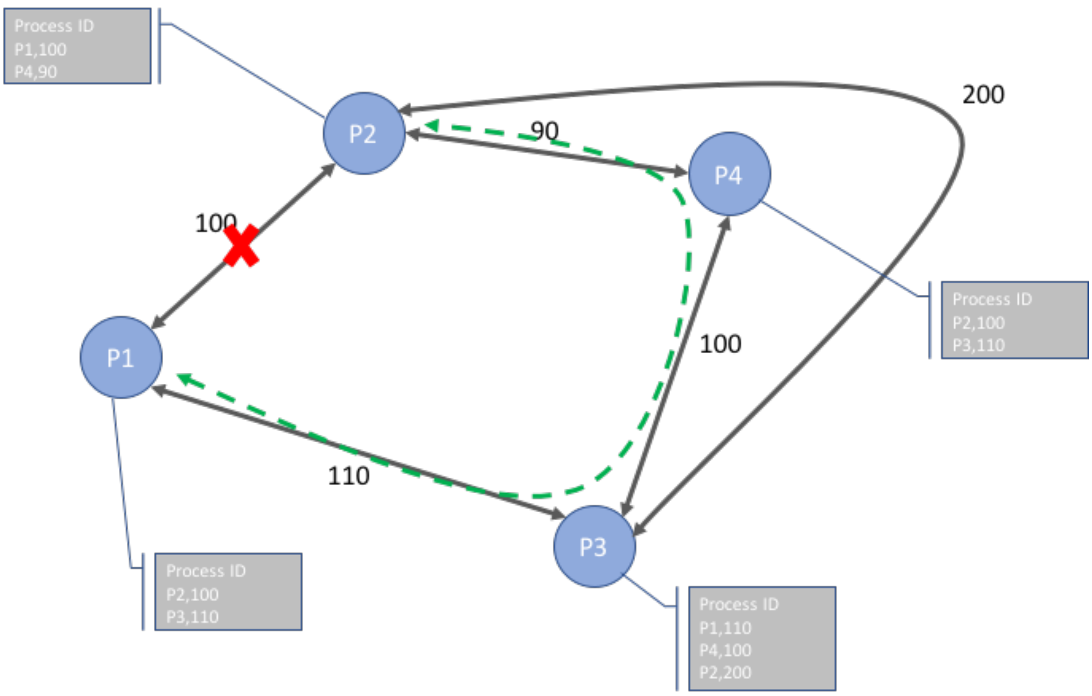


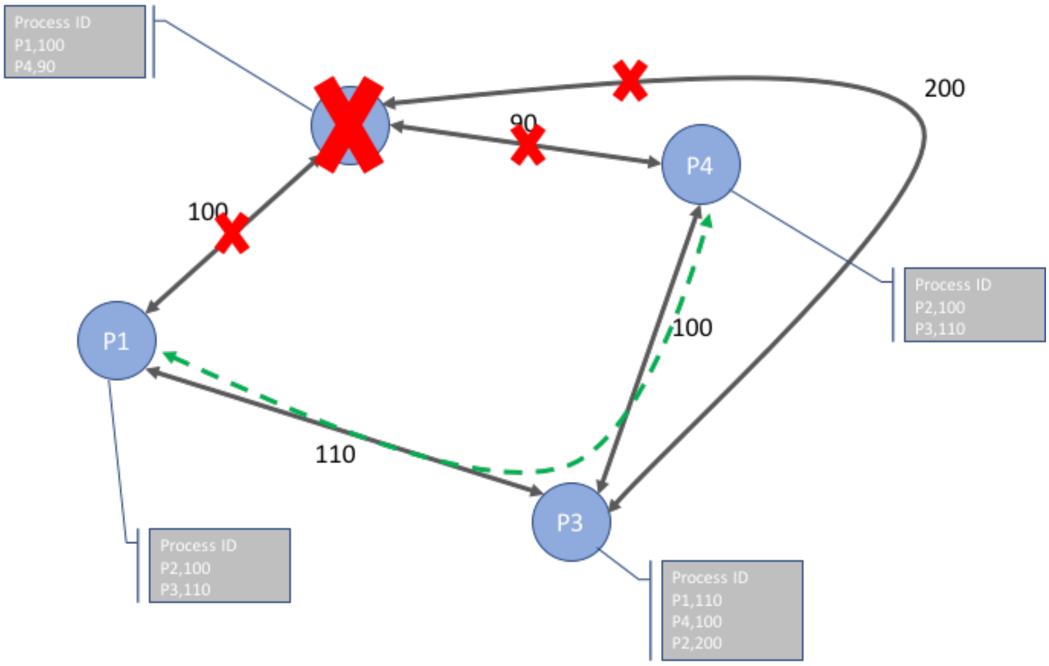
Problem Statement

It is required to build and simulate these processes locally as nodes. Once simulated, they should read the neighbor relationship file and identify the shortest path to reach from point A to point B.



In case of, edge or node failure the remaining nodes should be able to recalculate and find the most optimized path.





Given this information, following questions need to be answered.

1. Find the shortest path from any to any node given.
2. Find the minimum spanning tree.

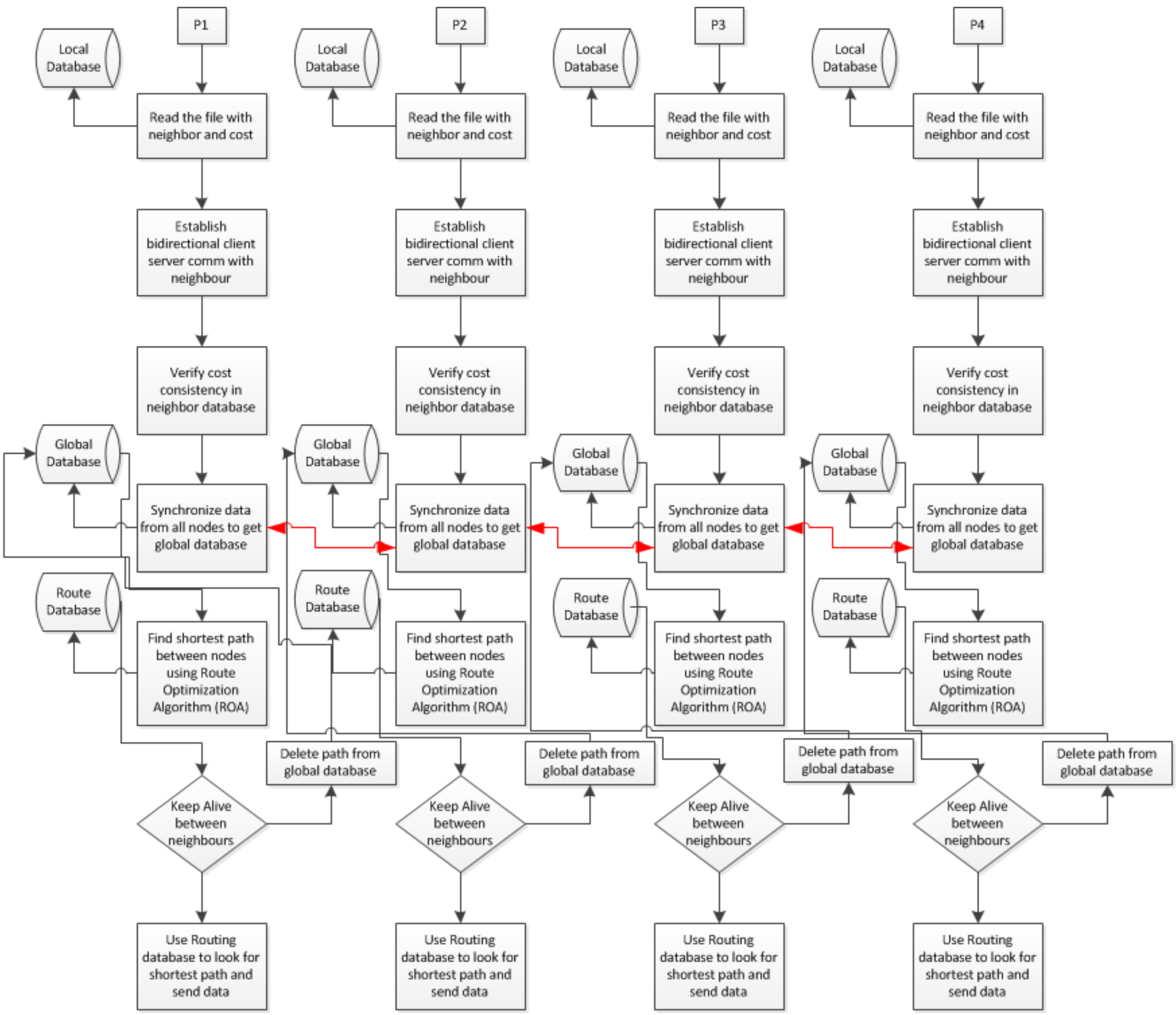
Also, reconfigure the network connectivity graph on a node when a node (process) dies or an edge is lost

Software Architecture

Code will be written in C language and all the nodes will be running as individual processes. A client-server communication will be used for transmitting data between all the nodes. Once the data is synchronized between nodes, a Routing Optimization Algorithm will be used to identify the shortest path and update it to routing database. All the processes need to send the keep alives among themselves to maintain the heartbeat. In case keep alives are missing then it will be considered that either edge or node is down. Text files will be used as neighboring database of nodes and there are three databases that are maintained in process memory:

1. Local Database
2. Global Database
3. Routing Database

Once the local database is read, the communication keeps on revolving between global database and routing database. One of the instruction as a part of hackathon is not to use any centralized mechanism hence none of the nodes is made to act as a master. All the nodes behave individually and programmed in same way to make sure that they provide same results. Complete architecture and data flow of program is as shown below.



External Functions

To get more information about the nodes and the way they behave, end user should be provided with some external functions. These functions can be used to understand the communication process between nodes and also in troubleshooting, if there are any issues. This program includes following functions.

ex.function 1: To show the shortest path between a source and destination node

ex.function 2: To view the complete topology of network

ex.function 3: To fetch routing database of each node

Testing

Test case	Scenario
1	All Nodes up
2	One of the edge is down
3	One of the node is down
4	One edge and one node down
5	Scale scenario for test case 2, 3 and 4 from minimum to maximum nodes
6	Cost consistency check
7	Availability of neighbor file
8	Errant process exclusion (DOS Attack)
9	Forced loop scenario to check spanning tree