We divided the implementation broadly into three files:

**1.      Prop\_node.py**

**2.      Network\_ctrl.py**

**3.      Client.py**

**Network.ctrl.py**

It contains the network implementation i.e it has the methods defined to send messages across different

servers based on their IP address. The important methods are listed below:

* **wait\_for\_ctrl\_connections** – This method is run in a background thread in each server so that servers can read messages from other servers.
* **send\_ctrl\_message\_with\_ACK** – This method has 5 parameters :
  + Message -  The data you want to pass.
  + messageType – What type of message it is.
  + extra – Any extra data you want to send.
  + requestNode – The node to which you want to send this message.
  + timeout – the timeout value

**Prop\_node.py**

This is the main file where the BFT Raft algorithm is implemented

Main methods:

* **join\_network** – Used by servers to join a cluster
* **start\_leader\_election** – Used by a server to start leader election
* **stabilization\_routine** – Used by servers to know about other servers in the cluster.
* **heartbeat\_routine** - As soon as a server becomes a leader it starts sending out heartbeat signals to existing nodes in the system to show that it is alive.
* **leader\_timeout\_routine** - When a node in the system does not receive heartbeat signal from the leader within its timeout period, it announces itself as a CANDIDATE and starts election by calling start\_leader\_election

**Client.py**

Client has **two major tasks.**

* Send commands to be executed to the server.
* If the command takes too long to get executed it intervenes to start a new leader election.