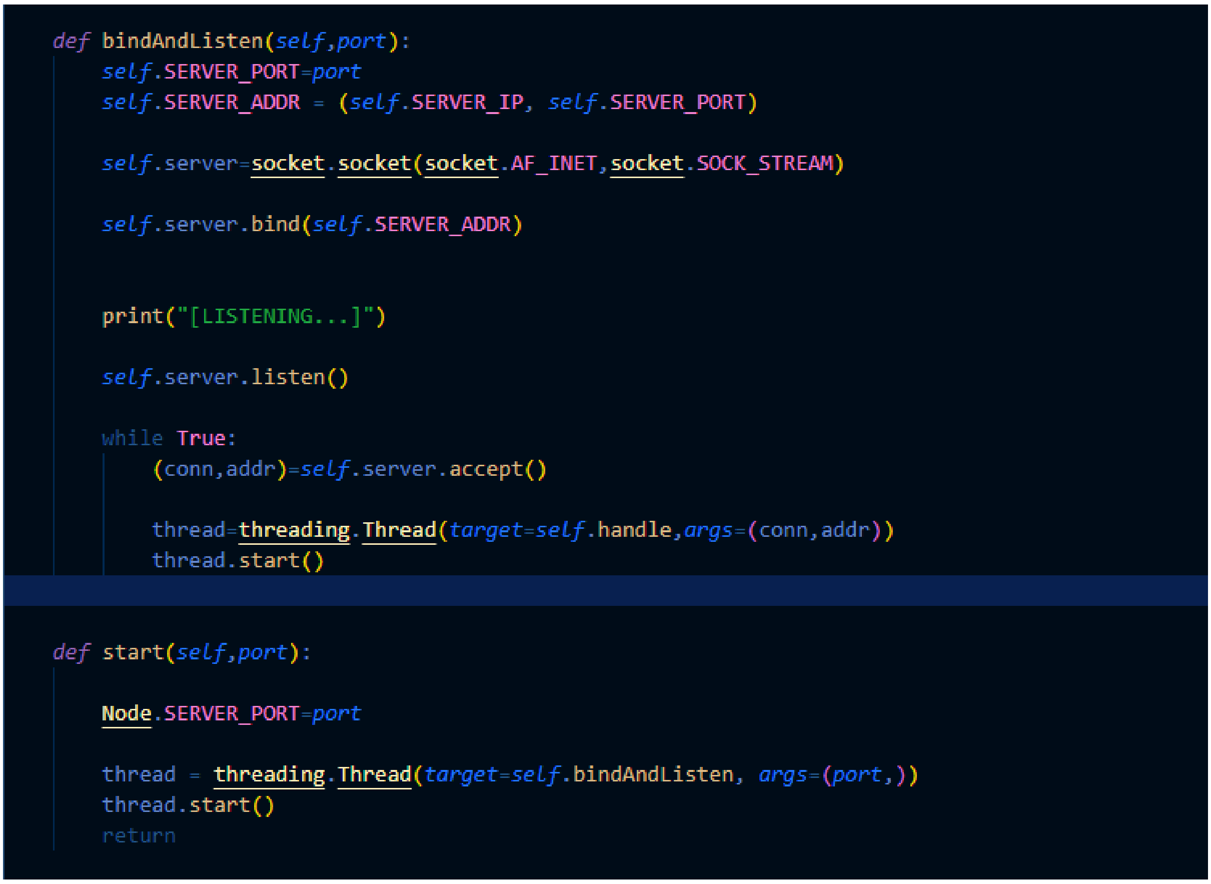
Overview

The included socket program implements a peer to peer network of nodes (hospitals) and allows for node-to-node communication as well as a BROADCAST message which sends a message to all nodes connected to the network. The protocol used is TCP/IP.

File Details

- **Settings and Helper directories:** Include modules which set up the environment.
- **commands.py:** specifies commands used in the main implementation
- **Message.py:** includes the definition for the message class as well as message object specifications.
- **Network.py:** contains code for the initialization and setup of the network, including the establishment of connections. The Network class has a variety of methods for this purpose.



```python
def bindAndListen(self,port):
    self.SERVER_PORT=port
    self.SERVER_ADDR = (self.SERVER_IP, self.SERVER_PORT)

    self.server=socket.socket(socket.AF_INET,socket.SOCK_STREAM)

    self.server.bind(self.SERVER_ADDR)


    print("[LISTENING...]")

    self.server.listen()

    while True:
        (conn,addr)=self.server.accept()

        thread=threading.Thread(target=self.handle,args=(conn,addr))
        thread.start()


def start(self,port):

    Node.SERVER_PORT=port

    thread = threading.Thread(target=self.bindAndListen, args=(port,))
    thread.start()
    return
```

The bindAndListen() and start() set up and initialise the network framework.

- **Node.py:** contains the Node class definition which assigns the head node address and port and creates node class parameters.

- **peer1.py, peer2.py… peern.py:** programs for each individual peer/node specifying addresses and ports for the nodes.

Instructions

Run/debug peer1.py on one machine, peer2.py on another, and so on. Follow instructions in the terminal environment for connection establishment.