B21CS063 B21CS031

Code description:

SEED:
1. Establish a socket, bind it, and then start listening for requests/messages.
2.Once it receives the connection request from the peer, it creates a separate thread.
3. When it receives a connection request from a peer, it sends its peer list to that peer.If a node receives a dead message from another node, it will be removed from its peer list.

PEER:
1. Retrieve seed address from connection file
2. Submit a request to the seeds.
3. Obtain a list of peers and establish connections with a selection of them at random.
4. It creates three separate threads: one for listening, another for gossip, and the last one for liveness testing.
5. It generates gossip messages, forwards them, and generates liveness requests.
6. When a peer is offline, it transmits a dead node message to the seed node.

Provide detailed instructions for compiling and running the programme.

1. Open the file "config.txt" and input the IP address and Port number in the "IP:PORT" format for the seeds.

2. We need to manually enter all Seed Addresses based on the number of seeds.

3. Access the files peer.cpp and seed.cpp in the terminal.(ubuntu)

To compile the seed, execute the code. Based on the number of entries in the configuration file, execute various seeds with different port numbers specified in the configuration file. Only input the port number when prompted manually. Now open multiple instances of the terminal using the same steps .

Compile peer.4. Execute the code by running peer.cpp. Choose any port number you prefer. Assign a unique port number to each peer. Proceed to run additional instances of the terminal following the same steps to create more peers.

Required commands for packages:

sudo apt update
sudo apt install openssl
sudo apt install libssl-dev

To compile peer:

1) gcc peer.cpp -o peer1 -lstdc++ -pthread -lssl -lcrypto

2) ./peer1

3) Replace peer1 with peer2 to generate another peer and run by ./peer2 , repeat this as required the peers

To compile seed:

4) gcc seed.cpp -o seed1 -lstdc++ -pthread

5) ./seed1

6) Replace seed1 with seed2 to generate another seed and run by ./seedr2 , repeat this as required the seeds(no, of entries in config file).