A Simple Peer To Peer Network Implementation

Hoora Abootalebi

Nariman Aryan

Amin Isaai

Amirhossein Khaje

Mahdis Tahdari

Ali Zeynali

November 2018

Contents

1 Introdeuction

This project aims to implement a peer to peer network.

2 Objects

First of all, we need to specify the objects in order to make the project more understandable and clear.

2.1 Streem

- **server**
- **nClient**:
- **dict(client:msg)** a dictionary to specify every client's message(s).
- **addClient(ip,port)**
- **removeClient(ip,port)**
- **randInBuff()**
- **parent**
- **sendMessage()**

2.2 Peer

- **stream()**
- **userInterface()**: Which the user or client sees and works with.

- **run()**: This method runs every time to see whether there is new messages or not.
- **packetFactory()**
- **handlePackets()**

2.3 Packet Factory

packetFactory would generate the packets every node needs to connect another with.

- **parseBuf()**
- **newReunion()**
- **newAdv()** make a new advertise packet.
- **newReg()** make a new register packet.

2.4 Packet

Every packet consists seven differntes parts: :**PlainText** which is the raw text message in the packet.
Node Sender Validator which make the packet valid.
Header where the information such as type of the packet and etc. are going to be there.
Body Action

2.5 Reunion

reunion(packet) checks the connection of the nodes to the root.

- **type**
- **getDest()**

2.6 Node

Every node has twp parameters: **IP** and **Port**.

2.7 Resgister Request

regReq() sends IP/Port of a node to the root to ask if it can register it.

2.8 Register Response

regRes() should just send an from the root *Ack* to inform a node that it has been registerd in the root if the regReq() was successful.

2.9 Advertise

adv(packet)

2.10 Mesasge

msg(packet)