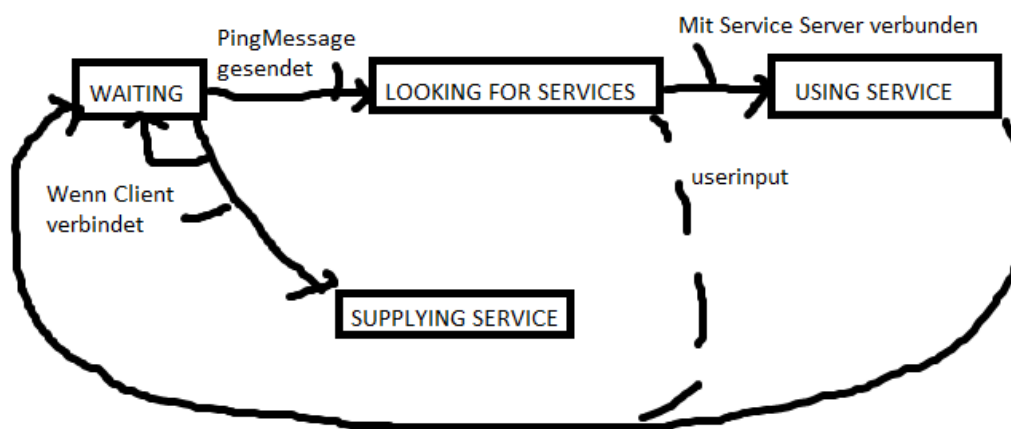


Protokoll:

Client öffnet ein DatagramSocket zum Senden
Wenn er Service anbietet öffnet er auch ein Socket zum lesen
Client schickt PingMessage als Broadcast
Services schicken PongMessage zurück
Client baut TCP-Verbindung mit gewünschten Service auf
ServiceOwner schickt StringMessage mit Antwort zurück

Zustandsdiagramm:

P2P User



Test:

```
Would you like to offer the Calculate Pi Service? [y | N ] y
Press enter to start looking for Services (this will take about 3 seconds)
```

```
The following Services where found:
0 | Pi Calculator
```

```
Please enter the ID of the service you want to use(or leave blank to not use any service): 0
PI is about 3.141592653589787
Would you like to discover services again? [y | N]
```

Ausführung:

In Windows über CLI mit `java -jar <jarname>` starten

Die Verwendung wird innerhalb des Programmes genauer erklärt

Antworten:

Mein Protokoll hat den Vorteil dass theoretisch durchgehen neue Services gesucht werden können. Ein starker Nachteil ist die starre Definition der Services, da diese keinen Input vom Client nehmen können und nur einen Output durchführen können. Dies musst aberso gemacht werden, um die Einfachheit zu gewährleisten.

Mit DatagramSockets können die Broadcasts sehr einfach implementiert werden. Dafür gibt es aber einige Einschränkungen, beispielsweise braucht ein Serviceanbieter so zwei Sockets anstatt nur einen, außerdem werden sich nie 2 P2PClients auf demselben Rechner befinden können, da UDPs zustandslose Verbindung in diesem Fall nicht mehr genau Sender / Empfänger finden könnte (Dies könnte durch Verbesserung des Protokolls behoben werden, aber wie schon oben erwähnt, verzichte ich auf zu komplexe Erweiterungen.

Weiteres:

Sollten sich auf dem PC des Benutzers aus irgend einem Grund mehrere Netzwerkinterfaces mit verschiedenen IP-Netzen befinden, kann dieses Programm unbestimmte Ergebnisse liefern.