

# **Vorlesung Rechnernetze**

## **Laborübung**

### **Server-basierter Peer-to-Peer Group-Chat**

**Prof. Dr. Dirk Staehle**

Die Abgabe erfolgt durch Hochladen der bearbeiteten Word-Datei in Moodle.

#### **Bearbeitung in Zweier-Teams**

**Team-Mitglied 1:**

**Team-Mitglied 2:**

## 1 Einleitung

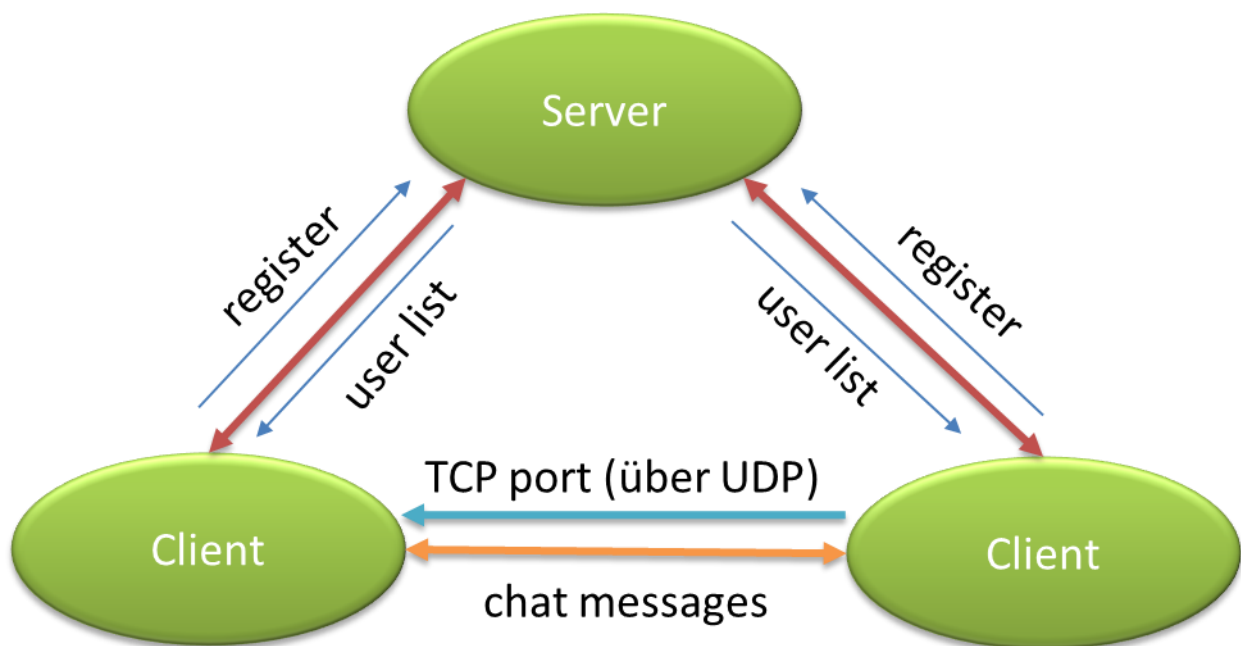
In dieser Aufgabe soll ein Anwendungsprotokoll für einen server-basierten Peer-to-Peer Group-Chat zunächst definiert und dann auf TCP- und UDP-Sockets implementiert werden.

## 2 Server-basierter Peer-to-Peer Group Chat

Das Ziel der Aufgabe ist, einen Group-Chat zu implementieren. Der Group-Chat besteht aus einem Server, der als zentrale Instanz die Verwaltung der Chat-Nutzer übernimmt und einer Menge von Chat-Peers, die untereinander chatten können.

Der Server wird zu Beginn gestartet und Chat-Peers können sich bei dem Server mit Nickname, IP Adresse und UDP Port registrieren. Der Server übermittelt dem Chat-Peer die Liste der angemeldeten Nutzer jeweils mit Nickname, IP-Adresse und UDP Port. Während der Client aktiv ist, erhält er vom Server auch Updates über neue Clients und Clients, die sich abgemeldet haben. Hier soll nicht die ganze Liste übertragen werden, sondern nur die Veränderung. Die Kommunikation zwischen Client und Server soll über TCP-Sockets stattfinden. Weiterhin soll der Server eine Broadcast-Funktion anbieten, d.h. ein Chat-Peer kann dem Server eine Nachricht schicken, die dieser an alle Chat-Peers weiterleitet.

Der Aufbau der TCP Verbindungen zum Peer-to-Peer Chat erfolgt über den Austausch der offenen Ports via UDP. Wenn ein Nutzer mit einem anderen Nutzer chatten will, zu dem noch keine Chat-Session besteht, muss zunächst eine TCP Verbindung aufgebaut werden. Dazu teilt der aktive Nutzer (Initiator) seinem Chat-Partner über UDP seinen offenen TCP Port mit. Der anvisierte Chat-Nutzer, baut dann eine TCP Verbindung zum Initiator auf. Diese TCP-Verbindung wird dann zum Austausch der Chat-Nachrichten genutzt.



### 3 Protokollspezifikation

Spezifizieren Sie Protokolle für den oben beschriebenen Group Chat, d.h. ein Protokoll für die Kommunikation zwischen Client und Server sowie ein Protokoll für den eigentlichen Chat zwischen den Clients.

Die Protokolle sollen die Syntax und Logik für den Group Chat festlegen. Das beinhaltet die Reihenfolge der ausgetauschten Nachrichten, die Nachrichtenformate und die Reaktion bei Fehlverhalten d.h. falsch formatierten, ausbleibenden oder unerwarteten Nachrichten.

Die Entwicklung der Protokolle soll als **Team übergreifende Arbeit** stattfinden. Dazu sollen sich mehrere (mindestens zwei) Teams zusammenfinden und ein **gemeinsames Protokoll festlegen und dann separat implementieren**.

Ziel diese Aufgabe ist, dass Sie lernen, was bei einem Protokoll festgelegt werden muss und welche Probleme auftreten, wenn einzelne Punkte nicht spezifiziert werden.

Achten Sie bei der Definition des Protokolls darauf, dass

- bei der Übertragung in TCP Nachrichten aus einem Bytestrom extrahiert werden können bzw. das Ende einer Nachricht erkennbar sein muss
- UDP Pakete verloren gehen können
- nicht nur der „good path“ sondern auch der „bad path“ spezifiziert werden muss, d.h. wenn statt der geplanten Nachricht keine, eine falsche oder eine unerwartete Nachricht eintrifft

### 4 Implementierung

Die Aufgabe besteht darin, dass jedes Team einen Server und einen Client/Peer implementiert. Die Peers und Server der verschiedenen Teams sollen untereinander kompatibel sein.

Testen Sie zunächst die Funktion ihrer eigenen Implementierung lokal auf ihrem Rechner und prüfen Sie in einem zweiten Schritt, ob ihre eigene Implementierung auch auf verschiedenen Rechnern funktioniert.

Nutzen Sie dazu z.B. die kostenfreie MeshNet-Funktionalität von NordVPN. Sie können sich entweder einen eigenen Account anlegen oder einen der Accounts `komnetze.htwg+x@gmail.com` (mit x von 1-8) verwenden. Das NordVPN-Passwort zu diesen Accounts ist jeweils `PasswortfuerdieRNVorlesung`. Wenn Sie sich auf verschiedenen Geräten mit dem Gleichen Account verbinden, sollten sich die Geräte im Gleichen MeshNet befinden. Testen Sie, ob sich die Geräte gegenseitig erreichen können (mit ping). Aktivieren Sie die Funktion Remote-Access, damit Sie auf Ports des Remote-Rechners zugreifen können (die Übung funktioniert). Sollte die Lösung mit NordVPN nicht funktionieren, können Sie das WLAN mit SSID `ubicomp` und Passwort `edisonGALILEO` nutzen.

## 5 Demo

Zur finalen Verifikation des Protokolls soll demonstriert werden, dass sich die Clients am Server des eigenen Teams und eines anderen Teams anmelden, miteinander chatten und sich dann wieder abmelden können.

Auch hier empfiehlt es sich, die Funktionalität zunächst lokal zu testen und dann in das VPN zu gehen.