

## Übungsblatt 7 – Multimedia im Netz

### **Aufgabe: Bidirektionale Audio-Kommunikation in Java**

Aufbauend auf die letzten Übungsaufgaben soll die bestehende Architektur zur Verbreitung von Audio-Daten über das Netz zu einer Lösung für „Voice over IP“ umgebaut werden. Die Kommunikation findet nun nicht mehr unidirektional (Producer – Server – Player, 1-zu-n), sondern bidirektional zwischen zwei Clients statt (Client zu Client, 1-zu-1) statt.

Auf dem Client werden Audio-Daten am Mikrofoneingang des Rechners aufgenommen und in Echtzeit an einen anderen Client geschickt, der diese unmittelbar abspielt. Eine Komprimierung z.B. als MP3 ist nicht nötig, verwenden Sie eine unkomprimierte Übermittlung in Mono mit niedriger Bitrate (z.B. 8 kHz bei 8 Bit pro Sample = 64 kbps). Die Gesamtverzögerung der Übertragung soll dabei möglichst gering sein, um eine Echtzeit-Sprachkommunikation zu ermöglichen.

Jeder Client öffnet für Steuerbefehle eine TCP-Verbindung zu einem anderen Client auf einem festgelegten Port (3333), die Audiodaten werden wie bisher über UDP übertragen (diesmal auf festgelegten Port 3333). Ein Client hat zu jedem Zeitpunkt maximal einen anderen Client als „Gesprächspartner“, es ist nicht nötig, den Datenstrom mehrerer anderer Clients abzuspielen oder den eigenen Datenstrom an mehrere Clients zu verschicken. Wie bei der vorherigen Aufgabe stellen Sequenznummern in den UDP-Paketen sicher, dass vertauschte Pakete erkannt und in der richtigen Reihenfolge abgespielt werden.

Um eine Verbindung zwischen zwei Clients herzustellen, öffnet der „anrufende“ Client eine TCP-Verbindung zur IP-Adresse des „anzurufenden“ Clients und signalisiert diesem den Gesprächswunsch (CONREQ). Dieser antwortet daraufhin zunächst mit einer Signalisierungs-Bestätigung (RING) oder einer Ablehnung (BUSY, wenn bereits ein Gespräch geführt wird bzw. NACK für alles andere). Die Gesprächswunsch-Signalisierung soll dabei visuell und akustisch erfolgen.

Nimmt der „angerufene“ Client das Gespräch an (ACK) und bestätigt daraufhin der „anrufende“ Client, dass er nun zur Übertragung bereit ist (OK), so kann diese auf beiden Seiten beginnen.

Beendet wird die Gesprächsanfrage oder das Gespräch entweder durch Schicken eines entsprechenden Kommandos über die TCP-Verbindung (CANCEL) oder bei Abbruch der Verbindung.

Die Implementierung soll in Java erfolgen und im Rechnerpool lauffähig sein. Die Lösungen müssen über das Online-Abgabesystem hochgeladen werden, neben Sourcecode und .class-Dateien soll eine Bedienungsanleitung enthalten sein. Diese Aufgabe darf in Gruppen zu je max. 5 Teilnehmern bearbeitet werden.

Abgabetermin: ~~Mittwoch, 9.02.2005~~

**Neuer Abgabetermin: Montag, 14.02.2005**