

Übungsblatt 6 – Multimedia im Netz

Aufgabe: Echtzeit-Audio-Streaming in Java

Wie bereits in voriger Aufgabe (Text-Streaming) soll von einem Producer ausgehend ein Echtzeit-Datenstrom über einen Server an mehrere verbundene Player verteilt werden.

Der Datenstrom soll diesmal allerdings aus Audiodaten bestehen, die bereits in komprimierter Form vorliegen, z.B. im Format mp3. Die Daten müssen allerdings nicht auch in Echtzeit generiert werden, sondern dürfen bereits als Dateien aus dem Dateisystem geladen werden. D.h. der Producer bietet die Auswahl einer oder mehrerer Dateien an, die dann allerdings als Datenstrom in Echtzeit (an den Server) weitergeschickt werden.

Echtzeit bedeutet hier, dass es der normalen Abspielgeschwindigkeit entsprechen muss, d.h. es wird im Endeffekt eine Live-Aufnahme simuliert.

Der Server wartet auf einem vordefinierten Upstream-Port auf UDP-Pakete von einem Producer (nur ein Producer pro Server ist erlaubt). Auf einem vordefinierten TCP-Port werden Player-Verbindungen registriert. Die TCP-Verbindung dient hierbei einem reinen Kommunikationskanal zur Steuerung ähnlich dem Protokoll „RTSP“, allerdings stark eingeschränkt: Ein Kommando zum Aushandeln eines UDP-Ports über den die eigentlichen Daten empfangen werden (z.B. SETUP <port>), und ein Kommando zum Starten der Datenübertragung (z.B. PLAY). Das Ende der Datenübertragung wird durch das Beenden der TCP-Verbindung veranlasst.

Der Player nimmt Kontakt zum Server über den bekannten TCP-Port auf und teilt diesem einen zufällig ausgewählten UDP-Port mit, über den er die Daten empfangen möchte. Danach wird dem Server mitgeteilt, er soll mit der Übertragung beginnen, und die empfangenen Daten werden dekomprimiert und abgespielt. Zum Abspielen (und Dekomprimieren) dürfen selbstverständlich frei verfügbare Bibliotheken verwendet werden.

Um Netzverzögerungen auszugleichen, soll der Player einen Puffer besitzen, der empfangene Daten vor dem Abspielen zwischenspeichert.

Da bei dieser Aufgabe zum Übertragen des Datenstroms UDP zum Einsatz kommt, ist darauf zu achten, die zu übertragenden Pakete mit Sequenznummern auszustatten, um die richtige Reihenfolge beim Empfang zu garantieren (ähnlich wie beim Protokoll „RTP“, wobei auf Zeitstempel und Quellenidentifikatoren hier verzichtet werden darf).

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.

Zwischenbesprechung: Donnerstag, 13.01.2005 **Abgabetermin: Mittwoch, 19.01.2005**