

Projektdokumentation - Rechnernetze & Kommunikationssystem

Clemens Dehnert

Januar, 2024

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Programm-Dokumentation	3
2.1	Funktionalität	3
2.2	Durchsatz in Theorie und Praxis	5
2.3	Probleme	6
2.4	Limitierungen	6
2.5	Verbesserungsvorschläge	6
3	Lernportfolio	7
3.1	Entwicklungsschritte	7
3.2	Lernfortschritte	7
3.3	Misserfolge	7

1 Aufgabenstellung

Im Rahmen des Moduls "Rechnernetze und Kommunikationssysteme" war ein Beleg zu erstellen, welcher ein Java-Programm zur Datenübertragung zwischen zwei Computern beinhaltet. Dabei war zusätzlich eine Dokumentation anzufertigen, wobei theoretisches Wissen mit der Praxis abgeglichen wurde.

2 Programm-Dokumentation

Dieses Kapitel behandelt, welche Funktionen das Java-Programm bietet und welche Funktionen nicht umgesetzt wurden.

2.1 Funktionalität

Das Programm bietet folgende grundlegende Funktionalitäten mit SW:

- Funktion Ihres Clients + Server ohne Fehlersimulation OK
- Funktion Ihres Clients + Server mit Fehlersimulation OK
- Funktion Ihres Clients + Server über Hochschulproxy (OK) Test über `./filetransfer Server 5555 100 0.2`
- Funktion Ihres Clients + Hochschulserver ohne Fehlersimulation OK
- Funktion Ihres Clients + Hochschulserver mit Fehlersimulation OK

Es wurden folgende Aufgaben der Aufgabenstellung mit SW umgesetzt:

- Aufruf des Clients OK
- Aufruf des Servers OK
- Datei wird beim Empfänger korrekt benannt OK
- Ausgabe Datenrate OK
- Implementierung Protokoll OK
- Fehlerkorrektur OK
- Implementierung Protokoll OK

Folgende Klassen und Methoden stehen zur Verfügung:

- FileCopy
 - `sendFile()`: initiiert Senden der Datei
 - `handleConnection()`: initiiert Warten auf Anfragen

- FileTransfer
 - file_req(): teilt zu sendende Datei auf und gibt Teile der Datei in SW
 - file_init(): setzt aus SW kommende Dateiteile zusammen
- SW
 - data_req(): sendet Daten aus FileTransfer
 - data_ind_req(): wartet auf eingehende Daten
 - closeConnection(): setzt Membervariablen der Servers zurück

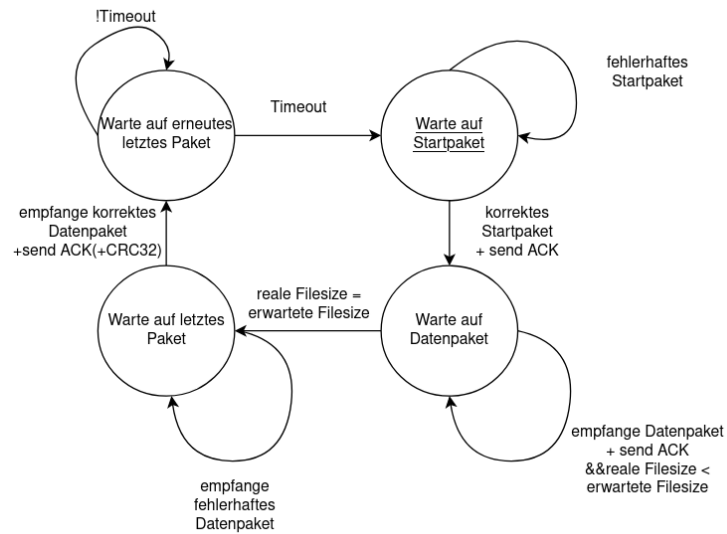


Abbildung 1: Zustandsdiagramm Server

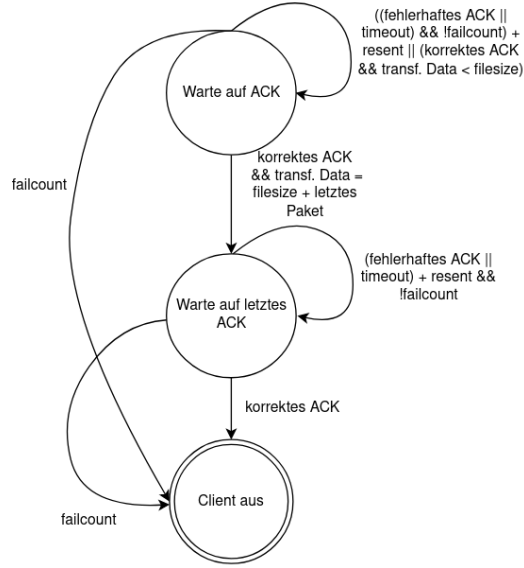


Abbildung 2: Zustandsdiagramm Client

2.2 Durchsatz in Theorie und Praxis

Der maximale theoretisch erzielbare Durchsatz bei 10 Prozent Paketverlust und 10 ms Verzögerung beträgt:

$$\eta_{\max} = \lim_{T_p \rightarrow \infty} \left(\frac{T_p}{T_p + 2 \cdot T_a + T_{\text{ACK}}} \cdot (1 - P_e)^2 \cdot R \right)$$

$$\eta_{\max} = \lim_{T_p \rightarrow \infty} \left(\frac{T_p}{T_p + 2 \cdot 0.01 + 0.01} \cdot (1 - 0.1)^2 \cdot \frac{1400}{1403} \right)$$

$$\eta_{\max} = 0.81$$

Der reale Durchsatz bei 10 Prozent Paketverlust und 10 ms Verzögerung beträgt:

$$\eta_{\text{real}} = \left(\frac{T_p}{T_p + 2 \cdot T_a + T_{\text{ACK}}} \cdot (1 - P_e)^2 \cdot R \right)$$

$$\eta_{\text{real}} = \left(\frac{T_p}{T_p + 2 \cdot 0.01 + 0.01} \cdot (1 - 0.1)^2 \cdot \frac{1400}{1403} \right)$$

$$T_p = \left(\frac{(1400 + 3) \cdot 8}{315000} \right)$$

$$\eta_{\text{real}} = 0.44$$

Die Datenrate ergab sich aus Messungen.

Der gemessene Durchsatz ist geringer, als der theoretisch maximale Durchsatz. Das kann damit begründet werden, dass die Timeouts bei verlorengegangenen Paketen nicht optimal gesetzt sind.

2.3 Probleme

Das Empfangen des CRC32 auf Seiten des Clients war kompliziert. Um der `waitForAck()`-Funktion in der Schicht SW zu signalisieren, dass sie das ACK-Paket des Servers mit dem CRC32 erhält, musste eine Membervariable verwendet war, wobei kein Boolean verfügbar war, wenn man nicht von der gegebenen Programmstruktur abweichen wollte.

2.4 Limitierungen

Es kann nur mit 'Stop and Wait' übertragen werden. Die Berechnung des Timeouts ist simpel, aber deshalb ist das Risiko, dass er ungünstig ist hoch.

2.5 Verbesserungsvorschläge

Eine detailliertere Beschreibung zur Verwendung des Hochschulservers und Proxys wäre hilfreich. Des Weiteren wäre eine höhere Gewichtung des Belegs motivierender.

3 Lernportfolio

Dieses Kapitel behandelt die persönlichen Lernfortschritte im Projekt.

3.1 Entwicklungsschritte

Zu Beginn des Projektes war ein grundlegender Aufbau des Programms bereits gegeben. Das war eine sehr hilfreiche Grundlage. Es war dennoch eine recht lange Einarbeitungszeit nötig, um die Klassen und das Arbeiten in Schichten zu verstehen. Es hat sich als sinnvoll herausgestellt, in kleinen Schritten vorzugehen. Das Protokoll SStop and Wait” wurde zuerst implementiert, wobei auf die Fehlerkorrektur zunächst verzichtet wurde. Als eine fehlerfreie Übertragung möglich war, wurde die Fehlerkorrektur implementiert, was sich als überraschend aufwendig herausstellte, da sich Bugs in dem mittlerweile etwas größerem Code nicht leicht finden ließen. Letztendlich wurden kleinere Features wie die Ausgabe der Datenrate oder die korrekte Dateibenennung für den Empfänger umgesetzt.

3.2 Lernfortschritte

Die bereits vorhandenen Kommentare zur Funktionalität einzelner Funktionsköpfen hat die konkrete Implementierung wesentlich erleichtert. Deshalb haben sich nur selten Fragen zum Algorithmus aufgetan - anders die Frage, wie sich etwas in Java konkret implementieren lässt. Da Java nur sehr kurz im Modul Programmieren 2 behandelt wurde, war insbesondere die Bytemanipulation und das Arbeiten mit Arrays nicht trivial, aber mit den richtigen Klassen nach etwas Einarbeitung möglich.

Des Weiteren war das Arbeiten in Schichten etwas Ungewohntes, da man prinzipiell nur den Umgang mit Objekten in einer objektorientierten Programmiersprache gelehrt bekommt.

Ein weiterer Lernaspekt war das Schreiben mit Latex. Hilfreich erwies sich dabei ChatGPT, was wesentlich schneller und präzise den passenden Befehl, den man sucht, liefert, als wenn man eine klassische Suchmaschine nutzt.

3.3 Misserfolge

Das Protokoll ”Go back N” wurde nicht umgesetzt. Grund dafür waren fehlende Zeit, aber auch Motivation, nachdem bereits eine mittlere zweistellige Anzahl an Stunden für Stop and wait aufgebracht wurde. Erschwerend kommt hinzu, dass der Beleg nur zu 30 Prozent in die Endnote eingeht. Nicht optimal lief noch, dass die Arbeit am Beleg die Zeit für die Vorbereitung der Seminare verringert hat. Sicherlich ist das eine Frage des Zeitmanagements, aber nach persönlichem Empfinden, erschwert die parallele Abarbeitung die Fokussierung wesentlich.