



**Vorlesung Rechnernetze**

**Theorieübung zu Page Load Times**

**Prof. Dr. Dirk Staehle**

Die Abgabe erfolgt durch Hochladen der Lösung in Moodle und exemplarisches Vorrechnen in der Laborübung.

**Bearbeitung in Zweier-Teams**

**Team-Mitglied 1:**

**Team-Mitglied 2:**

In dieser Aufgabe wird eine Web-Seite betrachtet, deren Aufbau in Tabelle 1 beschrieben ist. Hinweise zur Notation und Parameter der TCP-Verbindungen finden Sie in Tabelle 2.

1. Im Browser und auf dem Web-Server läuft die http-Version „Persistent http **ohne** Pipelining“. Skizzieren Sie in **Fehler! Verweisquelle konnte nicht gefunden werden.** den Verlauf der Seitenübertragung **beginnend mit dem ersten Request bis zur** **vollständigen Übertragung von Bild 5**. Tragen Sie pro RTT die Anzahl Segmente im Socket (Spalte Tx), die Sendefenstergröße (Spalte cwnd) sowie die Anzahl und den Inhalt der vom Server bzw. Client übertragenen Segmente ein. Gehen Sie davon aus, dass alle Pakete, die gleichzeitig gesendet werden auch gleichzeitig und in der richtigen Reihenfolge ankommen.
2. Im Browser und auf dem Web-Server läuft die http-Version „Persistent http **mit** Pipelining“.Skizzieren Sie in Abbildung 2 den Verlauf der Seitenübertragung **beginnend mit dem ersten Request bis zur** **vollständigen Übertragung von Bild 5**.
3. Im Browser und auf dem Web-Server läuft die http-Version „Persistent http **ohne** Pipelining“. Skizzieren Sie in Abbildung 3 die Übertragung von Bild 15 und Bild 16, wenn **Segment 56 von Bild 15 als erstes und einziges Segment bei der Übertragung der Web-Seite verloren** geht. Wie groß ist jeweils das Sendefenster des Web-Servers nach Empfang des letzten ACKs für ein Segment von Bild 15 bzw. von Bild 16?

Verwenden Sie folgende Notation, wobei Bytes die Anzahl der übertragenen Bytes pro Request bzw. Response ist.

* Requests: MOReq[Bytes], IO**n**Req[Bytes]
* Segmente der Objekte:
  + MO:S**x**[Start-Bytes]-S**y**[End-Bytes]
  + IO**n**:S**x**[Start-Bytes]-S**y**[End-Bytes]
    - Segment- und Byteangaben können bei vollständige Objekten bzw. Segmenten entfallen
* ACKs: n x ACK (die Anzahl genügt)

Tabelle 1: Aufbau der Web-Seite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objekt | Request | | Response (Objekt) | |
|  | Bytes | Segmente | Bytes | Segmente |
| HTML Code | 1200 | 2 | 2000 | 3 |
| Bild 1 | 1600 | 2 | 3200 | 4 |
| Bild 2 | 3200 | 4 | 800 | 1 |
| Bild 3 | 4800 | 6 | 8000 | 10 |
| Bild 4 | 800 | 2 | 30000 | 38 |
| Bild 5 | 1200 | 2 | 11600 | 15 |
| Bild 6-14 | 800 | 1 | 8000 | 10 |
| Bild 15 | 800 | 1 | 52000 | 65 |
| Bild 16 | 800 | 1 | 11200 | 14 |
| Bild 17 - 27 | 800 | 1 | 3200 | 4 |

Tabelle 2: Übersicht weiterer Parameter

|  |  |
| --- | --- |
| MSS | 800 Bytes |
| IW | 2 MSS |

Abbildung 1: Skizze für Persistent http ohne Pipelining

Client (Browser)

Server

cwnd

cwnd

Tx

Tx

Abbildung 2: Skizze für Persistent http mit Pipelining

Client (Browser)

Server

cwnd

cwnd

Tx

Tx

Client (Browser)

Server

cwnd

cwnd

Tx

Tx

Abbildung 3: Skizze für Persistent http ohne Pipelining mit Paketverlust