



## Kommunikationsnetze I – 2024

### Übungsblatt 8

#### Aufgabe 1

Vergleich am 21. Juni 2024

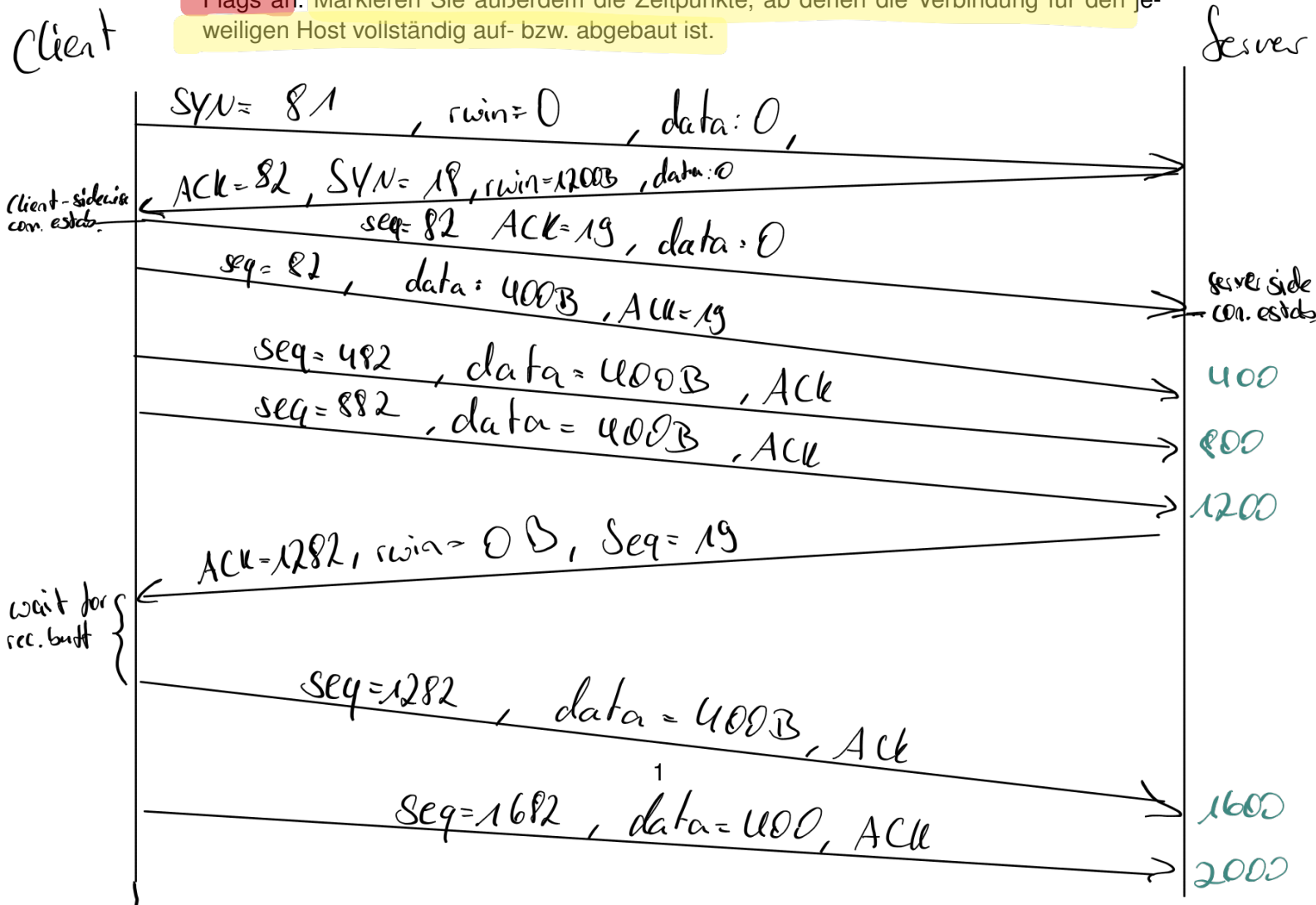
Ein Client verbindet sich mit einem Server und sendet 2000 Bytes über eine TCP-Verbindung. Der Client wählt als initiale Sequenznummer 81, der Server wählt die initiale Sequenznummer 18.

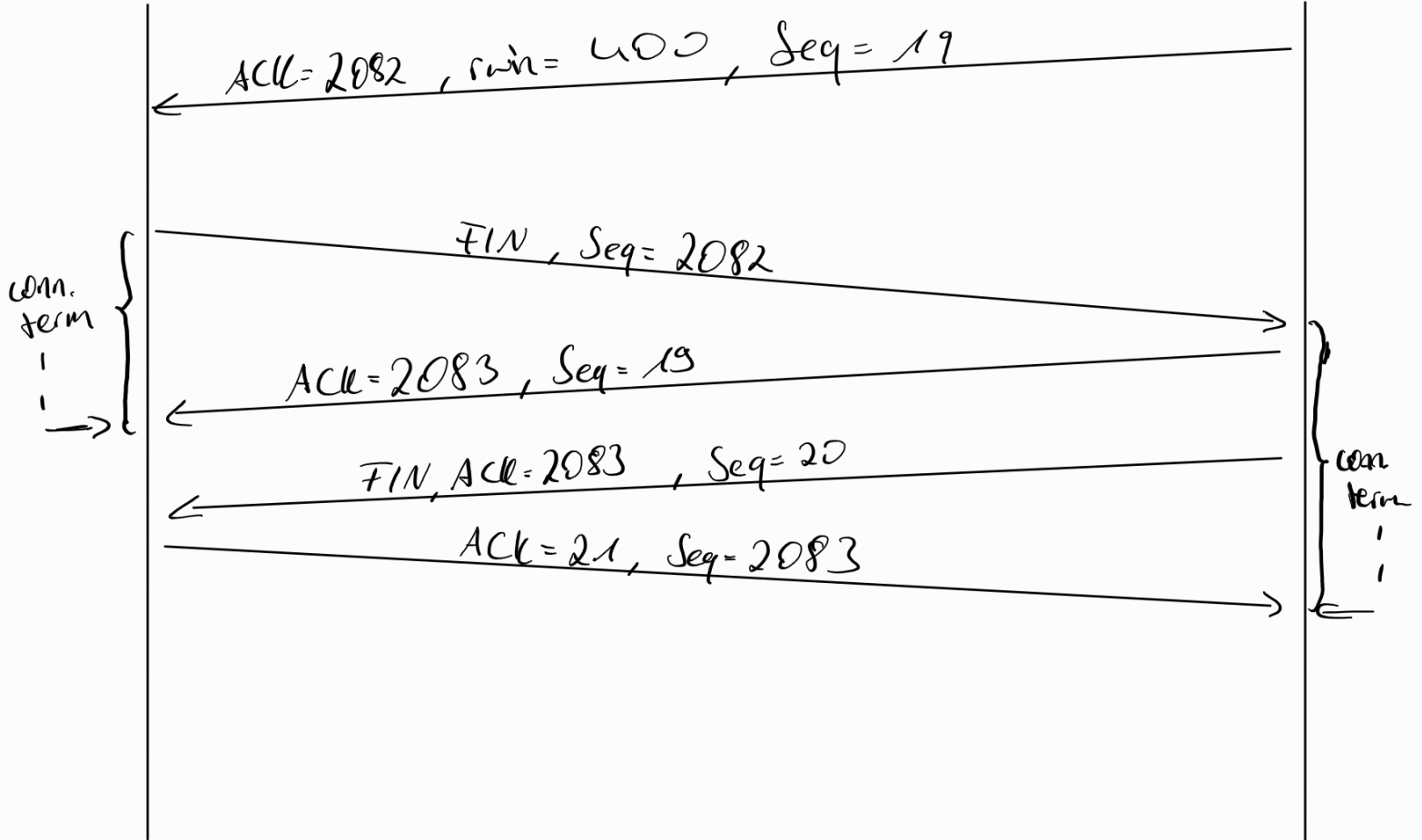
Das MSS ist konstant und beträgt 400 Bytes. Empfangene Daten werden sofort von der Anwendungsschicht gelesen. Der Empfangspuffer hat eine Grösse von 1200 Byte. Die Übertragungsdauer der Pakete kann vernachlässigt werden.

Der Empfänger nutzt Delayed ACKs. Es treten keine Paketverluste auf. Beschränkungen durch die Überlastkontrolle können Sie ignorieren. Die Verbindung wird nach Abschluss der Datenübertragung erst durch den Client, anschließend auch durch den Server geschlossen.

Zeichnen Sie ein Zeit-Ablauf-Diagramm, welches die komplette Lebensdauer dieser TCP-Verbindung darstellt. Geben Sie für jedes verschickte Segment die Werte des Sequenznummernfeldes (seq), des Bestätigungsfeldes (ack, soweit zutreffend), des Receive-Window-Feldes (rwin), die Anzahl enthaltener Nutzdatenbytes (data) sowie alle relevanten Flags an. Markieren Sie außerdem die Zeitpunkte, ab denen die Verbindung für den jeweiligen Host vollständig auf- bzw. abgebaut ist.

$w = R \cdot RTT$





**Bonusaufgabe 8 (6P)***Abgabe bis 21. Juni 2024, Vergleich am 21. Juni 2024*

Nehmen Sie an, Sie nutzen eine 2G-Mobilfunkverbindung um sich mit dem Internet zu verbinden. Sie ermöglicht eine Download-Datenrate von 240 kbit/s und eine Uploadrate von 100 kbit/s (12 500 Byte/s). Die RTT zu einem Webserver, mit dem Sie kommunizieren möchten, ist konstant und beträgt 30 ms. Es gehen keine Pakete verloren.

Nehmen Sie an, dass eine TCP-Verbindung zu dem Webserver bereits persistent etabliert ist, aber seit einiger Zeit nicht genutzt wurde. Alle Sende- und Empfangspuffer sind zunächst leer, alle versendeten Daten bereits erfolgreich bestätigt. Über diese Verbindung möchten Sie nun die unten stehende HTTP-Anfrage senden. Ihre Anwendung übergibt die Anfrage zeilenweise an die TCP-Schicht (das bedeutet: Es erfolgt ein separater Schreibaufruf für jede Zeile der Anfrage).

Die MSS Ihrer TCP-Verbindung ist bereits bekannt und beträgt 1400 Byte. Sie können außerdem annehmen, dass für jedes gesendete TCP-Segment insgesamt 40 Byte an Headerdaten anfallen. Die Übertragungsdauer der ACKs sowie die Effekte von Überlast- und Flusskontrolle können hierbei vernachlässigt werden.

Berechnen Sie für die folgenden Szenarien, wie lange es dauert bis die Anfrage komplett abgesendet ist, wie viele Daten insgesamt gesendet werden und wie hoch der verursachte Protokoll-Overhead ist (d. h., wie viel mehr Daten relativ zur Nutzdatenmenge versendet werden müssen).

- (a) Der Nagle-Algorithmus ist deaktiviert, sendebereite Daten werden jeweils sofort gesendet.
- (b) Der Nagle-Algorithmus ist aktiv.
- (c) Der Nagle-Algorithmus ist aktiv, der Server setzt Delayed ACKs ein und verzögert deshalb sein erstes ACK.
- (d) Die TCP-Schicht sendet alle Daten gleichzeitig, sobald die gesamte Anfrage vorliegt.

# Bytes	Inhalt
41	GET /teaching/current-courses/ HTTP/1.1\r\n
31	Host: www.kom.tu-darmstadt.de\r\n
94	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) [...]\r\n
95	Accept: text/html,application/xhtml+xml,application/xml; [...]\r\n
42	Accept-Language: de,en-US;q=0.7,en;q=0.3\r\n
36	Accept-Encoding: gzip, deflate, br\r\n
8	DNT: 1\r\n
24	Upgrade-Insecure-Requests: 1\r\n
30	Connection: keep-alive\r\n
26	Sec-Fetch-Dest: document\r\n
26	Sec-Fetch-Mode: navigate\r\n
22	Sec-Fetch-Site: none\r\n
20	Sec-Fetch-User: ?1\r\n
2	\r\n [Leerzeile, Ende des Headers]
497	

a)