

Übungsblatt 4

Kommunikationssysteme

Aufgabe 1: Es soll eine Datei der Größe L vom Host A zum Host B übertragen werden, wobei die MSS (Maximum Segment Size) 1460 Byte beträgt.

- Wie groß kann L maximal sein, damit keine TCP-Sequenznummer zweimal verwendet werden muss?
- Berechnen Sie für die eben ermittelte Dateigröße L die Übertragungsdauer. Nehmen Sie hierzu an, dass Transport-, Netzwerk- und Sicherungsschicht insgesamt 58 Byte Header zu jedem übertragenen Segment zufügt, bevor das resultierende Paket über eine Leitung mit 100 Mbit/s geschickt wird. Flusskontrolle und Überlastabwehrmechanismen sollen nicht berücksichtigt werden.

Aufgabe 2:

Host A und B haben eine TCP-Verbindung aufgebaut, Host B hat von Host A alle Bytes bis einschließlich dem Byte Nr. 358 erhalten. Host A sendet kurz hintereinander zwei Segmente an Host B, eines mit 50 Byte und eines mit 80 Byte. Das erste Byte des ersten Segments hat die Nummer 359, der Ursprungsport hat die Nummer 1028 und der Zielport hat die Nummer 80. Host B schickt für jedes von A empfangene Segment ein ACK und speichert jedes empfangene Segment.

- Geben Sie für das zweite von A gesendete Segment die Nummer von Ursprungsport und Zielport und die Sequenznummer an.
- Bei B kommt das zuerst gesendete Segment als erstes an. Geben Sie für das von B gesendete ACK-Segment den Ursprungsport, den Zielport und die ACK-Nummer an.
- Falls das zweite von A gesendete Segment als erstes bei B ankommt, welche ACK-Nummer sendet B, nachdem auch das erste von A gesendete Segment angekommen ist?
- Nehmen Sie nun an, dass die Segmente in der richtigen Reihenfolge übertragen werden. Weiter soll das erste ACK verloren gehen und das zweite ACK nach dem ersten Timeout von Host A ankommen. Weitere Verluste sollen nicht auftreten. Zeichnen Sie ein Diagramm mit dem zeitlichen Ablauf der Verbindung. Geben Sie für jedes Datensegment die Folgenummer und die Anzahl der Bytes und für jedes ACK-Segment die ACK-Nummer an.

Aufgabe 3: Verwaltung von Sende- und Empfangspuffer

- Ein Programm zum Dateitransfer hat seiner TCP-Instanz Datenoktetts im Folgenummernbereich 5718 ... 10717 übergeben. Die TCP-Instanz hat davon 4500 Datenoktetts gesendet und 4000 Datenoktetts quittiert bekommen.

Beschreiben Sie in einer Skizze den Zustand des Sendepuffers durch die Zeiger, *LastByteWritten*, *LastByteSent* und *LastByteAked*.

- b) Die entfernte TCP-Instanz hat Datenoktets in den Bereichen 5718 ... 7717 und 8001 ... 10100 empfangen. Die entfernte Anwendungsinstanz hat bis einschließlich Oktett 6600 alle empfangenen Daten gelesen.

Beschreiben Sie in einer Skizze den Zustand des Empfangspuffers durch die Zeiger, *LastByteRcvd*, *LastByteRead* und *NextByteExpected*.

- c) Welche Folgennummer referenziert *NextByteExpected*, nachdem das Segment mit den Oktetts 7718 ... 8000 (aber kein weiteres) in der entfernten TCP-Instanz eingetroffen ist?

Aufgabe 4: Die Fragmentierung von Datagrammen und das erneute Zusammensetzen wird von IP gehandhabt und ist für TCP unsichtbar. Heißt das, daß sich TCP über die Ankunft von Fragmenten in der falschen Reihenfolge keine Gedanken machen muss?

Aufgabe 5: Betrachten Sie die Anwendung der Slow-Start-Methode auf einer Leitung mit einer Rundreisezeit von 10ms. Überlastungen liegen nicht vor, das Empfangsfenster ist 24Kbyte, und die maximale Segmentgröße 2Kbyte. Wie lange dauert es, bis das erste volle Fenster übertragen werden kann?

Musterlösung

1)

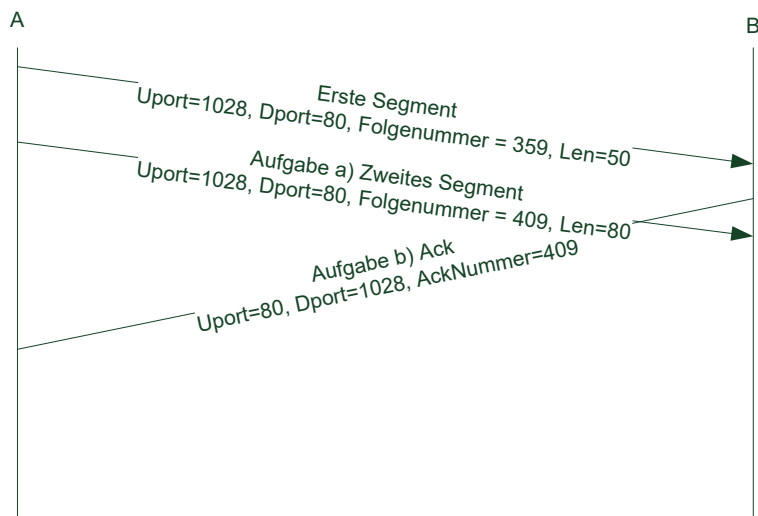
- a) Sequenznummer zählt die Anzahl der versendeten Nutz-Bytes. Es stehen 32 Bits zu deren Speicherung zur Verfügung. Es können somit $2^{32} = 4,29 \cdot 10^9$ Nutz-Bytes versendet werden.

$$L = 4,29 \cdot 10^9 \text{ Byte}$$

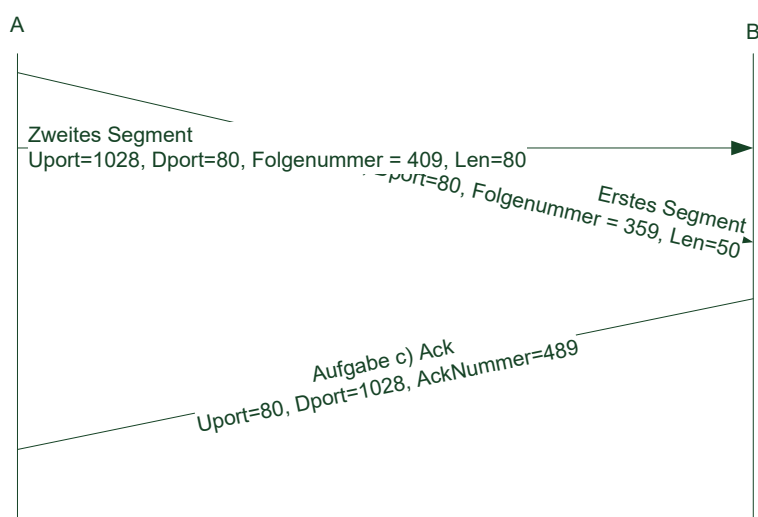
- b) $L_{\text{Brutto}} = (1460 + 58) / 1460 \cdot L = 4,46 \cdot 10^9 \text{ Byte}$

$$t = L_{\text{Brutto}} / R = 4,49 \cdot 10^9 \cdot 8 \text{ bit} / (100 \cdot 10^6 \text{ bit/s}) = 357 \text{ s} = 6 \text{ min}$$

2) a) und b)

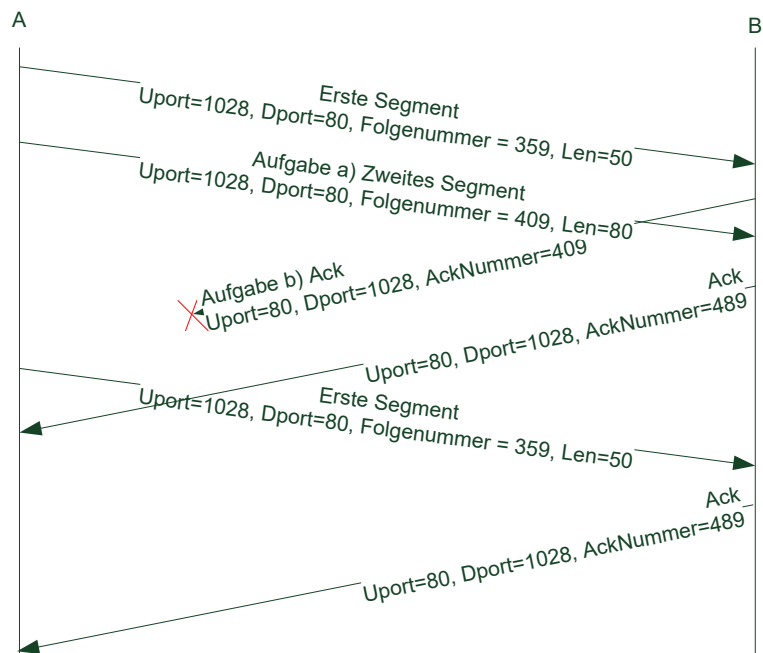


c)



Bem: Wegen der Übersichtlichkeit wurde das Ack des Zweiten Segmentes nicht eingezeichnet. Es hätte die AckNummer 359.

d)

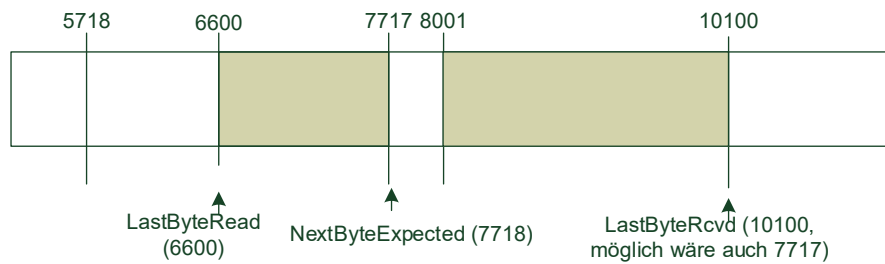


Aufgabe 3:

a) LastByteAcked bei 9717, LastByteSent bei 10217



b)



c) 10101

Aufgabe 4) TCP muss sich keine Gedanken über die Reihenfolge der Fragemente machen, sehr wohl aber um die Reihenfolge der zusammengebauten Datagramme.

Aufgabe 5: (Annahme: Linearer Anstieg bei Hälfte der Windowssize)

1 Runde: 1 Segment mit 2Kbyte

2 Runde: 2 Segmente mit je 2 Kbyte = 4 Kbyte

3 Runde: 4 Segmente mit je 2 Kbyte = 8 Kbyte

4 Runde: ~~8 Segmente mit je 2 Kbyte = 16 Kbyte~~ Hälfte Windowssize → 6 Segmente = 12 Kbyte (6,5 = 13Kbyte)

5 Runde: 7 Segmente mit je 2 Kbyte = 14 Kbyte (7,5 = 15Kbyte)

6 Runde: 16Kbyte (17Kbyte)

7 Runde: 18Kbyte (19Kbyte)

8 Runde: 20 Kbyte (21Kbyte)

9 Runde: 22 Kbyte (23Kbyte)

10 Runde: 24KByte

d.h. 90ms