

Rechnernetze, Übung 7

HENRY HAUSTEIN

Aufgabe 1

- (a) Die unteren Schichten schicken Daten nur an IP-Adressen, also an Rechner. Aber auf dem Rechner laufen viele Anwendungen, sodass das Betriebssystem keine Ahnung hat, für welche Anwendung die Daten bestimmt sind. Mit einem Port kann spezifiziert werden, für welche Anwendung die Daten gedacht sind. → UDP
- (b) Eine Anwendung weiß nicht, ob Pakete unterwegs verloren gegangen sind oder ob die Pakete in der richtigen Reihenfolge angekommen sind. Mit einer aufgebauten Verbindung lassen sich diese Probleme lösen. → TCP

Aufgabe 2

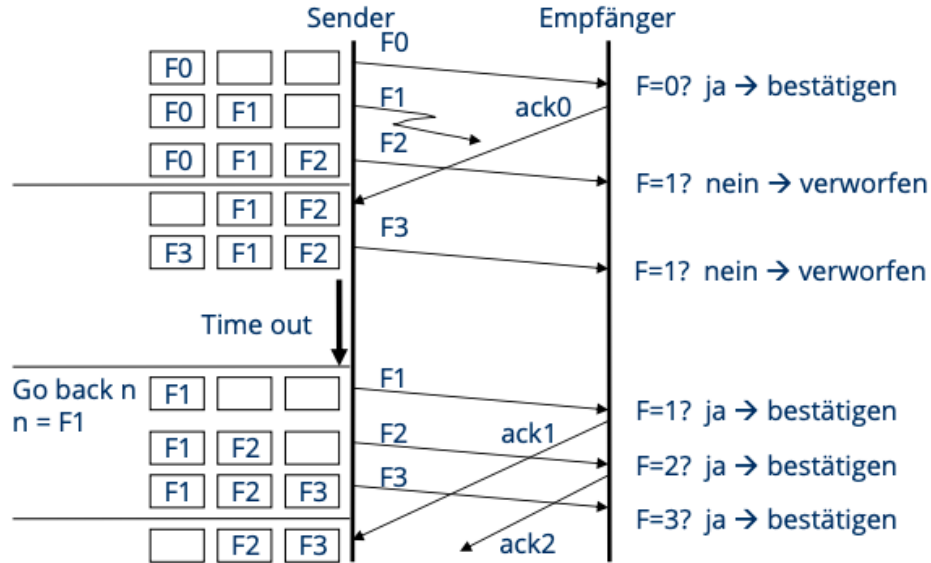
- (a) TCP: verbindungsorientiert, UDP: verbindungslos. Aber es gibt noch einige andere Unterschiede:

	TCP	UDP
Verbindungsverwaltung	Vollduplexverbindung, gepufferte byteweise Datenübertragung, Verbindungsaufbau mit 3-Wege-Handshake	nicht vorhanden
Reihenfolgegarantie	Ja (32-bit Folgenummern)	keine
Fehlerbehandlung	Erweitertes Prüfsummenverfahren (TCP-Pseudoheader; enthält Angaben aus TCP und IP-Header)	Einfache Prüfsumme (nur über die UDP-Pakete)
Flusskontrolle	Ja (variable Fenstergröße)	keine
Anwendungen	Große Datenvolumen mit Reihenfolgegarantie, Anwendungen ohne Echtzeitanforderungen	Multimediaanwendungen in schnellen + sicheren Netzen, zeitkritische Anwendungen

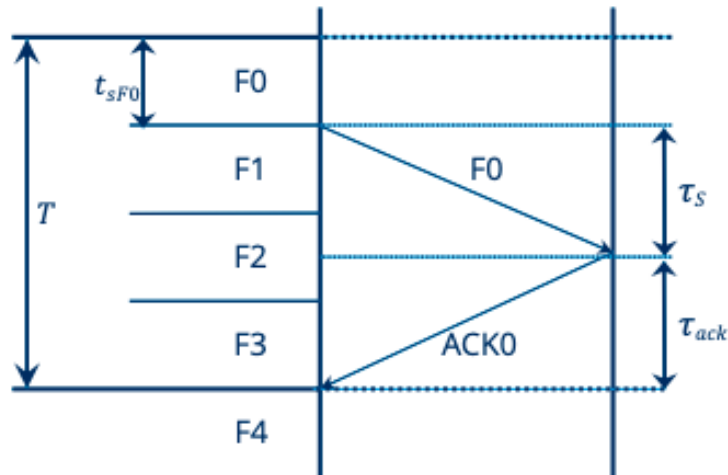
- (b) Eine Anwendung, die UDP nutzt, muss daher gegenüber verloren gegangenen und umsortierten Paketen unempfindlich sein oder selbst entsprechende Korrekturmaßnahmen beinhalten.

Aufgabe 3

(a) Ablauf

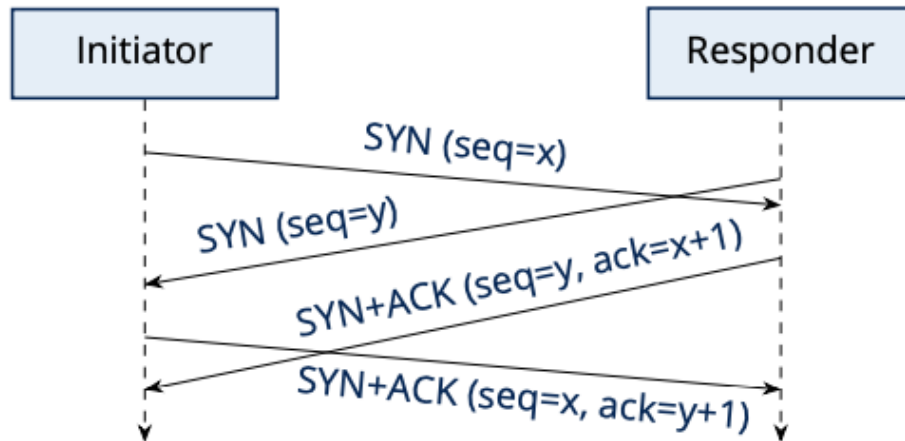


- (b) Sollten alle Bestätigungen zu F gesendeten Paketen verloren gehen, fängt der Sender nach einem Timeout an, diese Daten nochmal zu senden und beginnt bei Laufnummer 0. Der Empfänger weiß aber nicht, ob das Paket mit der Nummer 0, was er gerade erhält, davon kommt, dass die Bestätigung zu den letzten F Paketen nicht ankam und deswegen die Strategie *go-back-n* sagt, dass die letzten F Pakete nochmal geschickt werden müssen, oder ob die Laufnummer einfach nur das nächste Paket ist, was der Sender sendet und alle Bestätigungen angekommen sind.
- (c) 100 % Auslastung bedeutet, dass der 4. Frame gesendet wird, wenn ACK_0 ankommt. Das heißt, nachdem F_0 abgesendet wurde, muss das Paket zum Empfänger laufen und die Bestätigung zum Sender zurück, insgesamt 40 ms. Währenddessen können F_1 , F_2 und F_3 bereits auf den Weg gebracht werden. Während also F_0 in der Leitung ist, können $40 \text{ ms} \cdot 40 \text{ kBit/s} = 1600 \text{ Bit}$ geschickt werden. Diese 1600 Bit entsprechen aber 3 Frames, also ist ein Frame $F = \frac{4000}{3} \text{ Bit} = 534 \text{ Bit}$ groß.

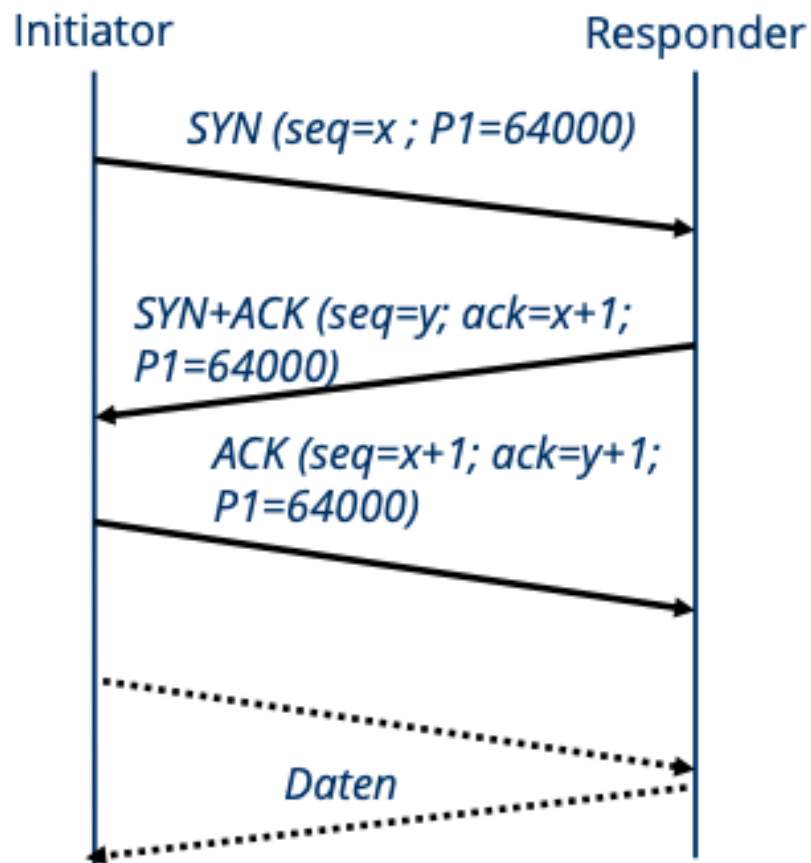


Aufgabe 4

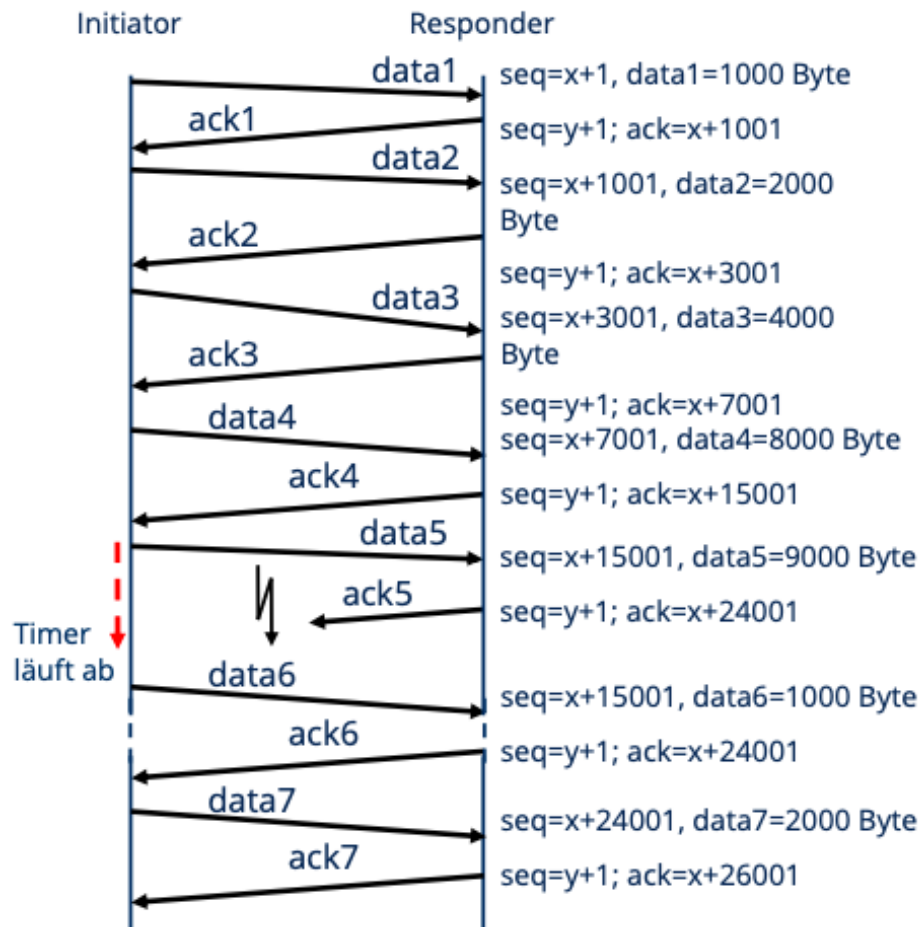
(a) Ablauf



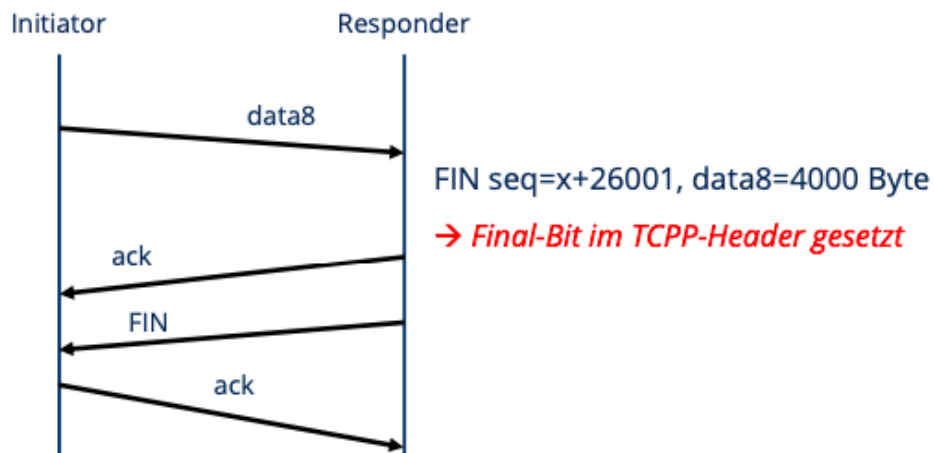
(b) $P1=64000 \rightarrow$ gewünschte Puffergröße 64000 Byte, seq: gesendete Bytes eines Datenstroms, ack: nächste erwartete Bytes, das SYN-Segment gilt als 1 Byte des Datenstroms, deshalb stehen die Zähler nach dem Verbindungsaufbau auf $x+1$ und $y+1$



Datenübertragung

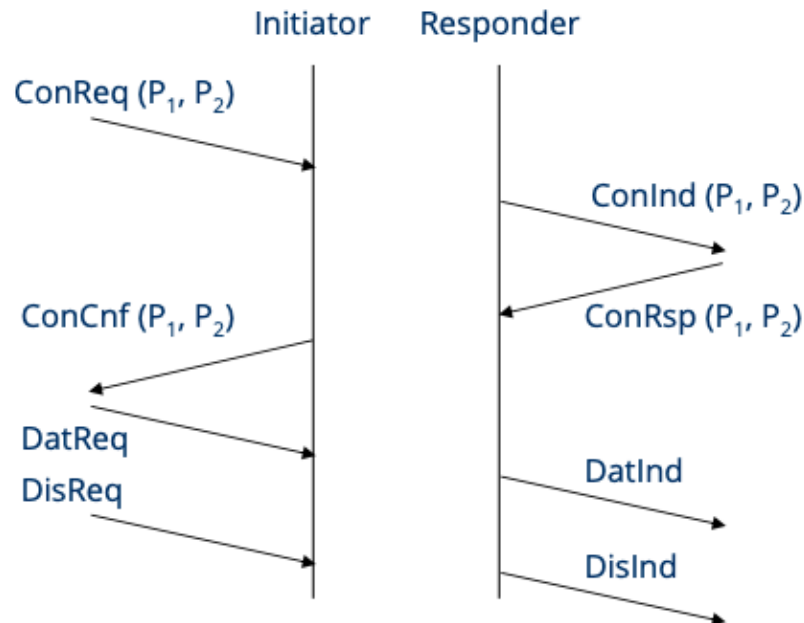


Verbindungsabbau



Aufgabe 5

(a) P_1, P_2, \dots, P_N sind Parameter, hier $P_1 = 1200 \text{ Byte/s}$, $P_2 = \text{verschlüsselt}$



(b) Responder ändert Datenrate auf 75 % → Initiator akzeptiert → Verbindung existiert zu reduzierter Datenrate

Responder ändert Datenrate auf 75 % → Initiator akzeptiert nicht → Initiator baut Verbindung ab
 Responder kann geforderte Dienstgüte (Verschlüsselung) nicht erbringen → Abbruch der Verbindung

