



Übung 10: Zuverlässige Datenübertragung, TCP

Aufgabe 1: Go-Back-N (GBN) und Selective Repeat (SR)

Es gilt das **fiktive**¹ Protokoll Version 3.0 der Vorlesung. Ggfs. ist die Animation interessant:
http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/

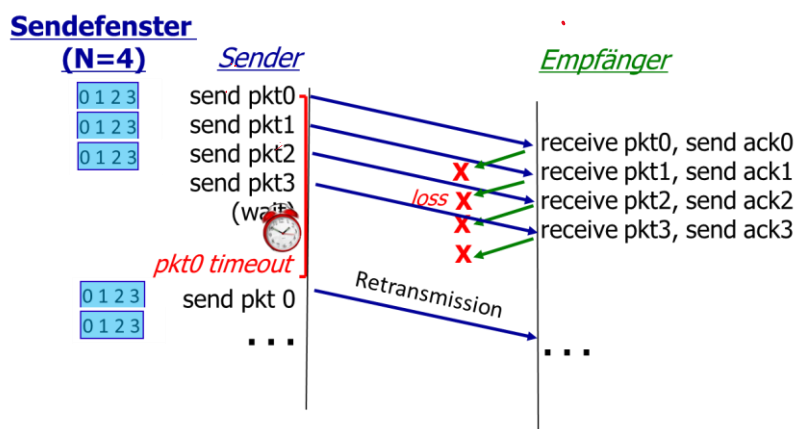
- Zwei Hosts *A* und *B* verwenden **GBN**². Die Sendefenster beträgt $N = 3$. *A* sendet 6 Nachrichten mit Payload zu *B* so schnell als möglich. Alle Nachrichten werden korrekt empfangen und unmittelbar bestätigt. Nur das 1. Acknowledgment von *B* zu *A* und das 5. Datensegment von *A* zu *B* gehen verloren. Zeichnen Sie ein Sequenzdiagramm!
- Es gilt a). Nun wird aber **SR** eingesetzt. Zeichnen Sie erneut ein Sequenzdiagramm! Was sind die Unterschiede zu a)?.
- GBN**: Welche Sequenznummern könnte das **Sendefenster** theoretisch enthalten?
 - Die Größe des Sendefensters ist $N = 4$.
 - Die Sequenznummern laufen von 0 bis 1024.
 - Der Empfänger benötigt als **nächstes** das Paket mit der Sequenznummer 500.

Aufgabe 2: Maximale Größe des Sende- und Empfangsfensters

Nehmen Sie an, dass für die Sequenznummern m Bits zur Verfügung stehen. Es gibt 2^m mögliche Sequenznummern, diese liegen im Bereich $[0, 2^m - 1]$. Im konkreten Beispiel für $m = 2$ sind das $\{0, 1, 2, 3\}$. Im Folgenden überlegen wir, wie groß das Sendefenster N bei **GBN** und **SR** sein darf.

Hilfreich ist die Betrachtung des Falles, wenn immer alle Acknowledgments verloren gehen.

- Was ist m beim realen TCP?
- GBN**: Das Sendefenster ist zunächst $N = 2^m = 4$, siehe Skizze. Warum funktioniert das nicht? Wo liegt das Problem?



- GBN**: Die Größe des Sendefensters sei $N = 2^m - 1 = 3$. Funktioniert das? Warum?
- SR**: Was passiert, falls $N = 2^m - 1 = 3$? Erklären Sie am Beispiel! Zeichnung!
Hinweis: Bei SR akzeptiert der Empfänger nur bestimmte Sequenznummern (Receive Window, siehe Folie 23 „Sicht des Empfängers“)

¹ Es werden fiktive Lehrprotokolle betrachtet, nicht TCP. Sequenznummern beziehen sich auf Pakete.

² Annahmen: Der Datenkanal ändert nicht die Reihenfolge der Pakete. Die Anzahl der verfügbaren Sequenznummern sei ausreichend.

Aufgabe 3: TCP – Unidirektionale Kommunikation

Host A und B kommunizieren über eine bestehende TCP Verbindung. B hat von A bereits **alle** Bytes bis einschließlich Byte 126 korrekt empfangen und bereits erfolgreich an A bestätigt. Nun schickt A direkt hintereinander 2 Segmente an Host B (*Segment 1* und *Segment 2*), ohne nach dem Senden von *Segment 1* auf das dazugehörige Acknowledgement zu warten:

- *Segment 1*: 80 Bytes Nutzdaten, Sequenznummer 127, Src Port 302, Dst Port 80
- *Segment 2*: 40 Bytes an Nutzdaten

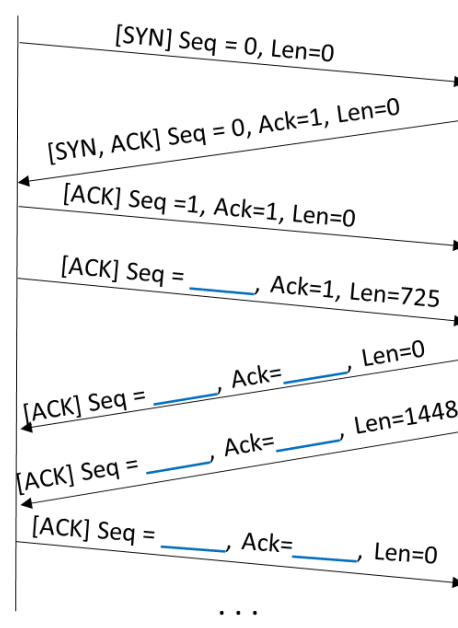
Annahme: Host B sendet immer sofort ein ACK, sobald ein Segment von A empfangen wurde.

- Was ist die Sequenznummer, der Source- und der Destination-Port im *Segment 2*?
- Szenario 1: *Segment 1* kommt vor *Segment 2* bei B an. Was ist die ACK-Nummer, Source- und Destination-Port im Acknowledgement, das B unmittelbar nach Erhalt von *Segment 1* an A sendet?
- Szenario 2: *Segment 2* kommt vor dem *Segment 1* bei B an. Was ist die ACK-Nummer im Acknowledgement, das B unmittelbar nach Erhalt von *Segments 2* an A schickt?
- Es gilt wieder Szenario 1: Das ACK für *Segment 1* geht auf dem Weg zu A verloren, während das ACK für *Segment 2* erst *nach* dem TCP Timeout des *Segment 1* bei A ankommt. **Zeichnen** Sie ein Sequenzdiagramm! Es tritt kein weiterer Paketverlust auf.

Aufgabe 4: TCP – Bidirektionale Kommunikation

Das rechte Sequenzdiagramm zeigt den Beginn einer TCP Verbindung. Es gibt keine Paketverluste oder Retransmissions. Beide Empfänger bestätigen den Empfang von Nutzdaten immer mit der nächsten Nachricht. Das Feld *Len* gibt für jedes Paket die Größe der mitgesendeten Nutzdaten in Bytes an.

Ergänzen Sie die 7 fehlenden SEQ-/ACK-Nummern an den **blau** gekennzeichneten Stellen. Die Zuweisung muss Sinn ergeben und den obigen Annahmen genügen.



Frohe Weihnachten und ein gutes neues Jahr!