



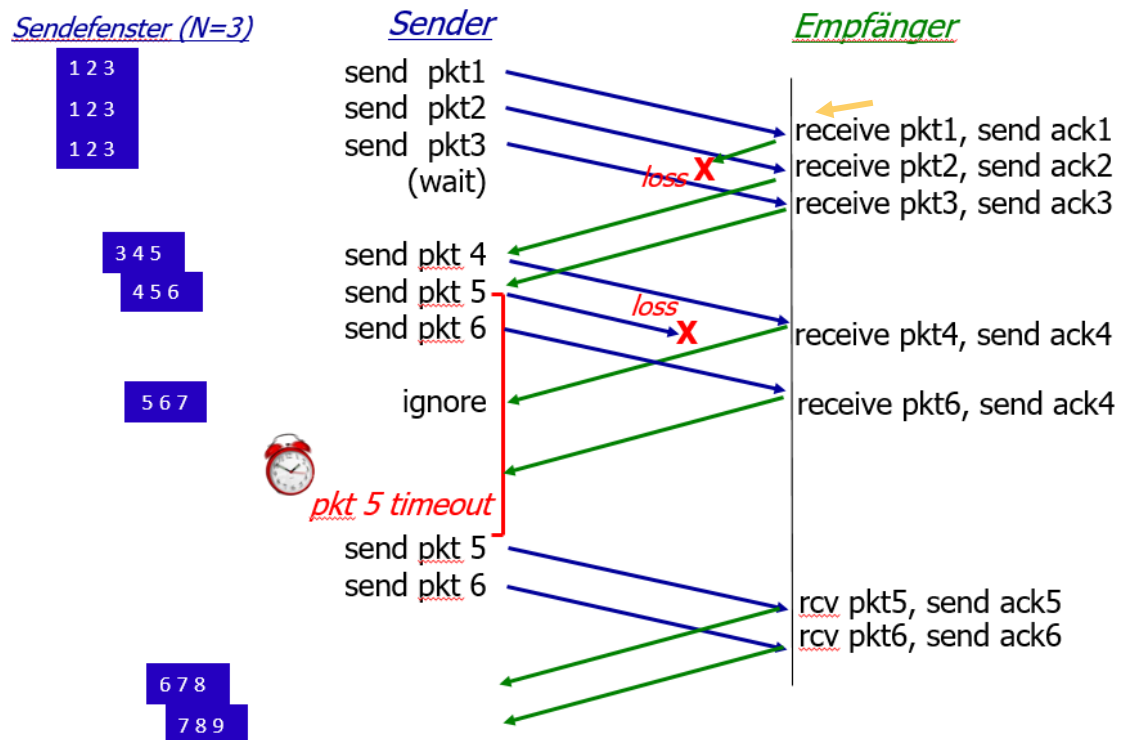
Lösung 10: Zuverlässige Datenübertragung, TCP

Aufgabe 1: Go-Back-N, Selective Repeat

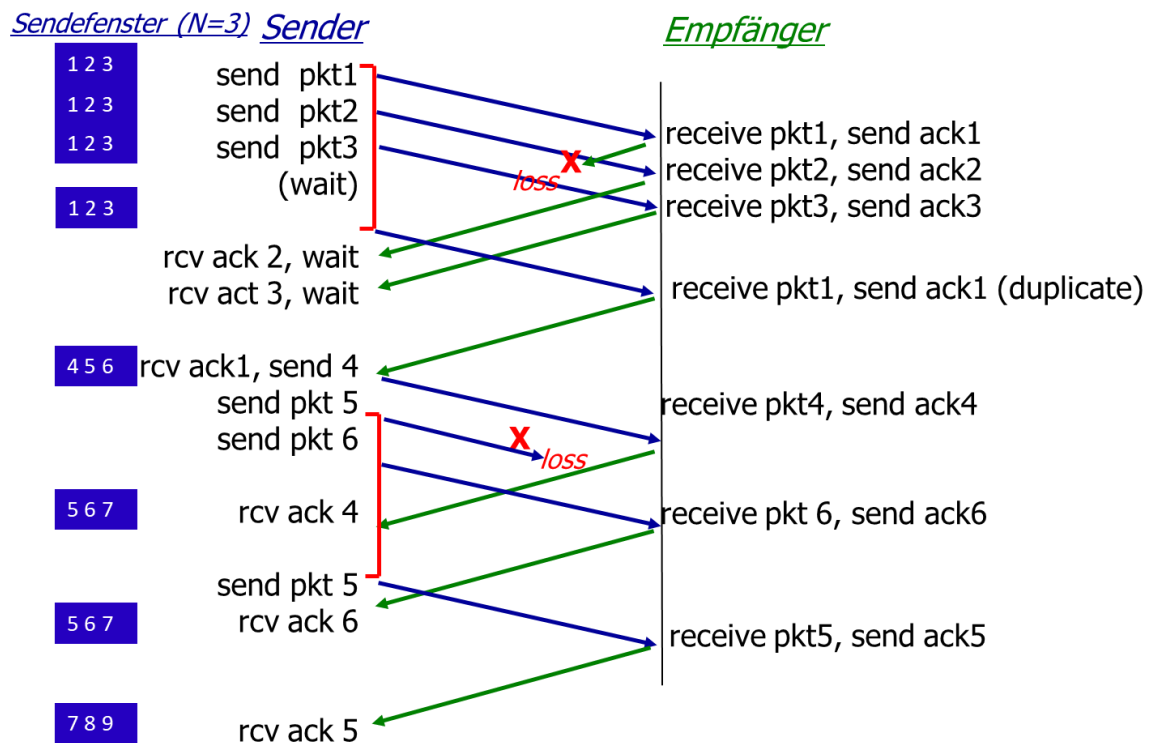
Hinweis: Im fiktiven Protokoll Version 3.0 der Vorlesung bestätigt ein ACK x immer, dass der Empfänger bereits alle Pakete bis einschließlich Sequenznummer x erhalten hat. Später bei TCP ist es leicht anders! Hier werden nicht Pakete sondern Bytes aus dem Bytestrom bestätigt. Ein ACK x bestätigt bei TCP, dass alle Bytes mit der „Nummer“ $< x$ bereits erhalten wurde, das nächste erwartete Byte ist x.

- a) Siehe Diagramm. Die Annahme hier ist, dass das ACK=2 beim Sender eintrifft, bevor der Timeout für Paket 1 ausläuft. Andere Annahmen sind erlaubt, solange die Gesamtlösung schlüssig ist.

Go-Back-N: Kumulative ACKs



- b) Vergleicht man die Diagramme, erkennt man 2 Unterschiede: Das 1. Segment muss jetzt erneut übertragen werden, das war vorher wegen der kumulativen Acknowledgments besser. Am Schluss dagegen muss nur das verloren gegangene Paket erneut übertragen werden. Weitere Details siehe Diagramm.



- c) Die Tatsache, dass der Empfänger als nächstes das Paket mit der Sequenznummer 500 erwartet bedeutet, dass er bereits das Paket 499 erhalten und bestätigt hat, sowie alle vorangegangenen Pakete. Nun ist nur nicht klar, ob der Sender die ACKs vom Empfänger für Paket 499, 498, 497, und 496 auch bereits erhalten hat. Das ACK für 495 muss er erhalten haben, ansonsten hätte der Sender das Datenpaket 499 wegen der Größe 4 des Sendefensters gar nicht an den Empfänger senden dürfen. Man unterscheidet am besten 2 Fälle:

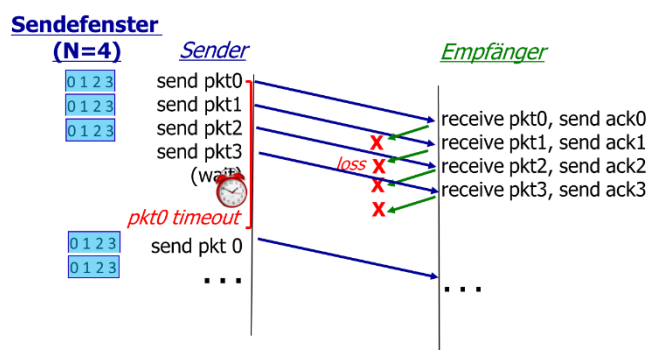
- Alle ACKs 499, 498, 497, 496 sind auch beim Sender eingetroffen: Dann ist das Sendefenster [500; 503].
- Keine der ACKs ist beim Empfänger eingetroffen: Dann ist das Sendefenster [496; 499]

Dies sind die beiden Extremfälle. Irgendwo „dazwischen“ kann das Sendefenster liegen.

Aufgabe 2: Maximale Größe des Sende- und Empfangsfensters

- a) Bei TCP stehen 32 Bit zur Verfügung. Die maximale Sequenznummer ist $2^{32} - 1 = 4294967295$, die minimale Sequenznummer ist 0.
- b) Das zeitlich gesehen 5. Paket („send pkt0“) ist eine **Retransmission**. Diese kann der Empfänger aber nicht als solche erkennen. Er erwartet als nächstes pkt4, das aber wegen des recht kleinen Sequenznummernraumes wieder die Sequenznummer 0 haben müsste.

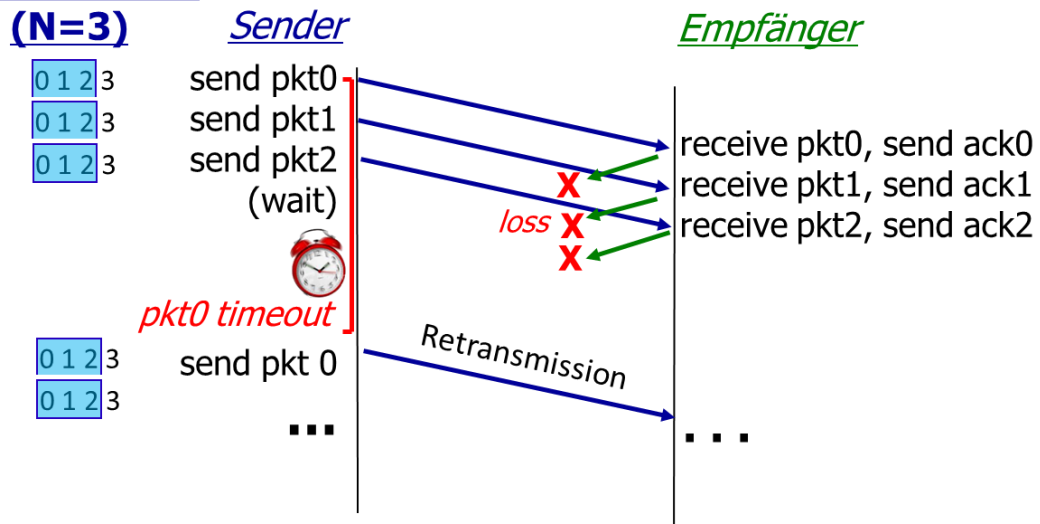
Im schlimmsten Fall gibt er die Daten als **Duplikat** im Betriebssystem an die Anwendung nach oben.



- c) Ja, das funktioniert! Der Empfänger erwartet das pkt3, siehe Bildchen. Es kommt aber erneut pkt0. Der Empfänger kann die Retransmission als solche erkennen!

Fazit: Allgemein gilt, dass die Größe des Sendefensters maximal $N = 2^m - 1$ sein darf, wenn m Bits für die Sequenznummern zur Verfügung stehen. Kleiner als dieser Wert N ist natürlich immer möglich, nur können dann weniger Pakete gleichzeitig unterwegs sein (Performance!)

Sendefenster

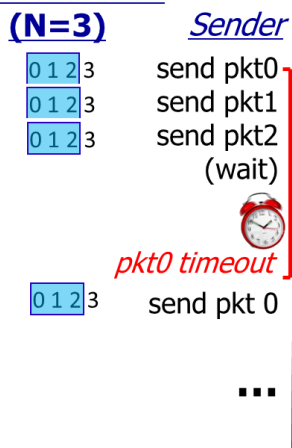


- d) In der Grafik ist illustriert, wie das Empfängerfenster liegt. Nur die blau markierten Sequenznummern sind aus Sicht des Empfängers gültig und werden von diesem akzeptiert und in Empfangspuffer gespeichert bzw. über das Betriebssystem weiter nach oben gegeben. Man erkennt, dass der Empfänger nach Erhalt von pkt2 die Sequenznummern 3, 0 und 1 akzeptiert und erwartet.

Trifft nun das erneut übertragene Paket nach dem Timeout ein, so hat dieses die Sequenznummer 0. Der Empfänger denkt, dass es ein neues Paket sei und eben keine Retransmission.

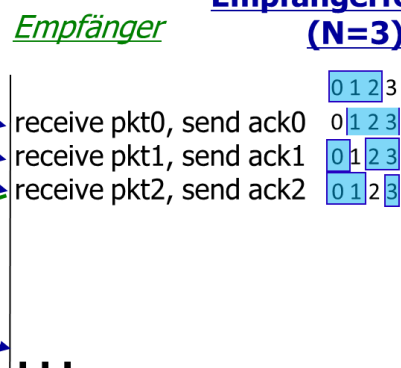
Fazit: Das Sende- bzw. Empfangsfenster ist für Selective Repeat zu klein. Bei Selective Repeat gelten strengere Anforderungen als bei Go-Back-N.

Sendefenster

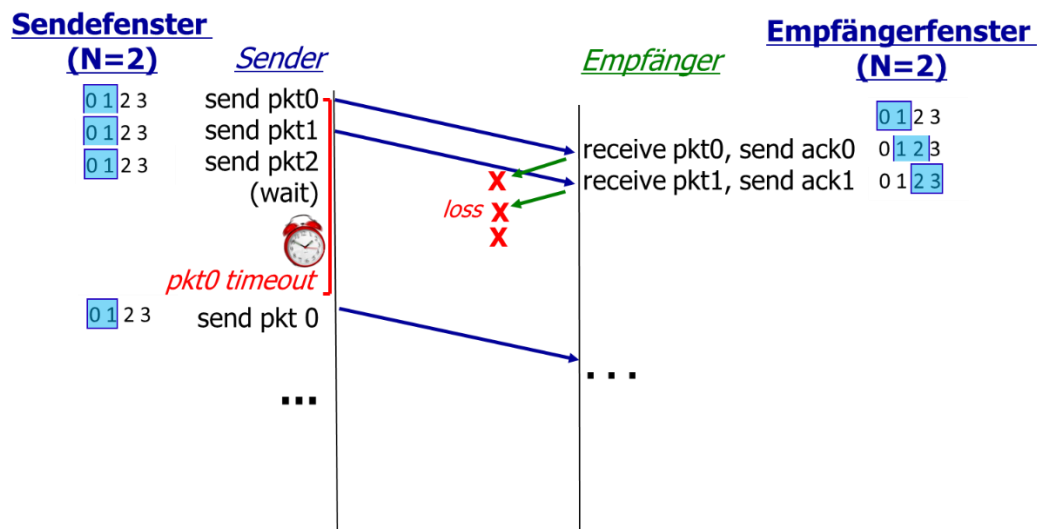


Empfängerfenster

(N=3)



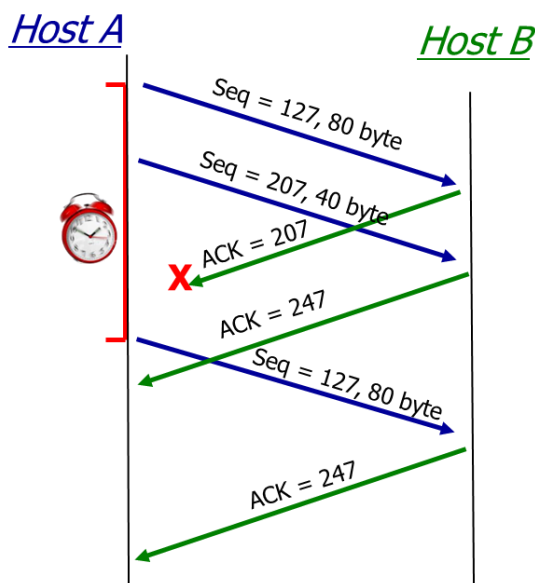
Weiterführend: Für die maximale Größe des Sendefensters bzw. Empfängerfensters bei Selective Repeat gilt: $N = 2^{m-1}$. In Worten: Das Sendefenster darf höchstens **halb so groß** sein wie die Anzahl verfügbarer Sequenznummern.



Aufgabe 3: TCP – Unidirektionale Kommunikation

- Seq = 207, Src = 302, Dst = 80
- Src = 80, Dst = 302, Ack = 207 (nächstes erwartetes Byte)
- Ack 127 (kumulative ACKs; als nächstes wird Byte Nummer 127 erwartet).
- Siehe Skizze!

Hinweis: TCP ist angelehnt an das Go-Back-N Prinzip. Im Gegensatz zum reinen Go-Back-N Ansatz wird bei einem Timeout aber nur das älteste noch unbestätigte Paket erneut gesendet.



Aufgabe 4: TCP – Bidirektionale Kommunikation

Hinweise:

- Beim Handshake wird für jede Richtung eine Sequenznummer verbraucht, auch wenn keine Nutzdaten fließen. Beispiel: Die 3. Nachricht trägt die Sequenznummer 1.
- Sequenznummern in einer Richtung „entsprechen“ den ACK-Nummern in der Gegenrichtung. Beispiel: ACK=726 in der vorletzten Nachricht und Seq = 726 in der letzten Nachricht.
- Die Sequenznummer einer Nachricht ergibt sich aus der Sequenznummer der Vorgängernachricht + Länge der Vorgängernachricht.

Beispiel: Die Seq=726 der letzten Nachricht ergeben sich aus der Seq=1 der 4. Nachricht + deren Länge 725 (Len = 725).

