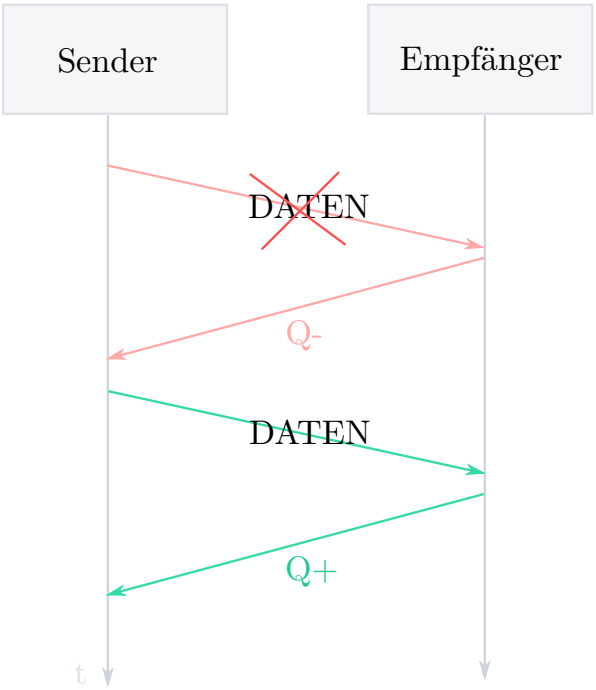
Kommunikationstechnik: Einführung?

R. Grünert
12. März 2021

1 Aufgaben von Protokollen

1.1 Aufgabe 1: Fehlerkontrolle

Bei Datenübertragungen können Fehler auftreten. Diese Fehlerfälle müssen erkannt und behandelt werden. Die erste Maßnahme ist, jedes Datenpaket mit einer *Quittung* zu bestätigen (positive Quittung) oder einen Fehlerfall zu kennzeichnen (negative Quittung).



Die Daten werden solange am Sender bereitgehalten, bis eine positive Quittierung zurückkommt. Erscheint eine negative Quittierung, können die Daten dann einfach erneut gesendet werden.

Wenn die Quittung selbst bei der Übertragung verfälscht wird, können zwei Fälle auftreten:

1. Daten gehen verloren aber man erhält eine positive Quittung
2. Daten werden empfangen, aber man erhält eine negative Quittung

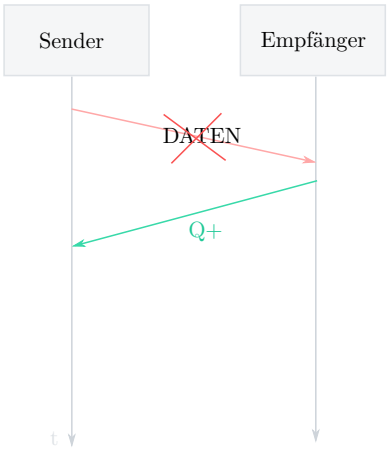


Abbildung 1: Pos. Quittung trotz Datenverlust

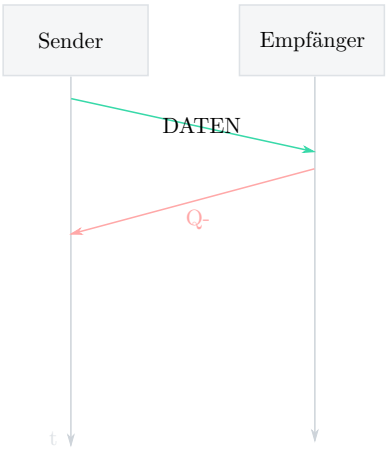
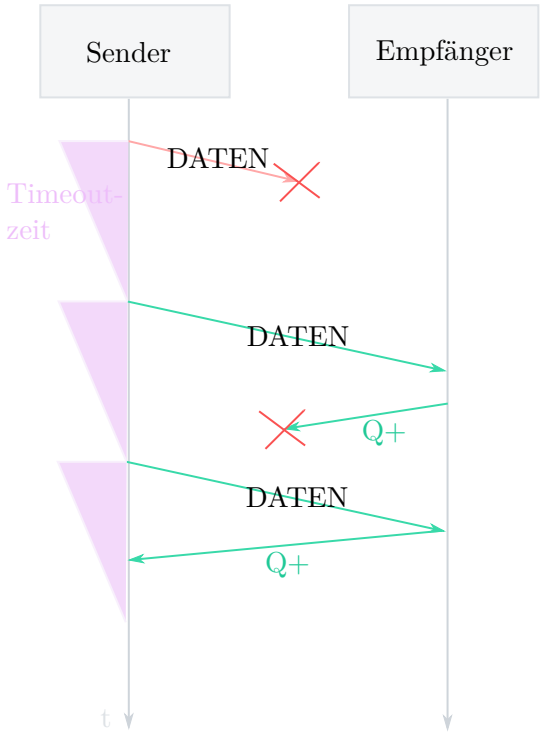


Abbildung 2: Neg. Quittung trotz richtiger Übertragung

Im ersten Fall kann es also dazu kommen, dass dem Empfänger Daten fehlen, diese Daten aber nicht erneut gesendet werden können, da der Sender eine positive Quittung erhielt und die vorherigen Daten bereits entfernt hat. Um dieses Problem zu beheben, müssen auch für Datenblöcke oder auch ganze Dateien Quittierungen gesendet werden, sodass diese im zweifelsfall komplett erneut gesendet werden.

Im zweiten Fall würde der Empfänger ein Datenpaket doppelt erhalten, da der Sender es aufgrund der empfangenen negativen Quittung erneut sendet. Um die Datenverdopplung zu umgehen, wird jedes Datenpaket nummeriert. Der Empfänger zählt einfach hoch und wenn die Nummer des empfangenen Paketes der momentanen Zählernummer entspricht, wird es weggeworfen. Natürlich kann man nicht unendlich hochzählen, daher werden Modulo-Zählverfahren, also einfach Rücksetzungen nach einer bestimmten Anzahl, verwendet (z.B. Modulo-8 oder Modulo-128).

Neben der Verfälschung von Daten oder Quittungen können auch **Verluste** auftreten, z.B. können Daten oder Quittungen aufgrund einer beschädigten Leitung oder eines verfälschten Headers nicht ankommen. Daher gibt es nur eine *begrenzte Zeit*, die maximal auf eine Quittung gewartet wird (Timeout). Nach dem Timeout kann der Empfänger die Daten dann erneut senden.



Es gibt zwei „ Eiserne Regeln“:

Eiserne Regel Nr. 1

Datenblöcke können **verfälscht** werden. Deshalb muss nach dem Absenden eines Datenblockes, dieser im Speicher gehalten werden für den Fall, dass eine erneute Übertragung notwendig ist.

Eiserne Regel Nr. 2

Datenblöcke können **verloren** gehen. Deshalb sollte man nur eine begrenzte Zeit (Timeout) auf die Quittung des Datenblockes warten.

1.2 Aufgabe 2: Flusskontrolle

Oft werden Daten beim Sender schneller produziert als sie vom Empfänger verarbeitet werden können. Der Sender muss dann seine Übertragung auf die Aufnahmefähigkeit des Empfängers anpassen.

Die Einfachste Lösung ist, dass der Empfänger dem Sender mitteilt, wann er keine Daten mehr aufnehmen kann (Halt) und wann er wieder in der Lage ist, welche aufzunehmen (Weitersenden). Das Problem hierbei ist, dass wenn einer dieser Befehle verfälscht wird oder verloren geht, die Übertragung fortgeführt oder komplett angehalten wird.

Eine weitere Lösung ist, dass der Empfänger dem Sender *Credits* verteilt, die er dann zum Senden von Datenblöcken aufbrauchen kann. Ein Credit definiert die Anzahl an Datenblöcken, die der Sender senden kann, ohne auf eine Quittung warten zu müssen. Auch hier muss darauf geachtet werden, dass der Sender die Credit-Mitteilungen erhält bzw. dass er keine doppelten Credits erhält.

Noch eine Möglichkeit ist der *Fenstermechanismus*. Vor der Übertragung sprechen sich Sender und Empfänger über eine *Fensterbreite* innerhalb des Wertebereiches der Sequenznummern (Nummerierung der Pakete) ab. Diese Breite sei W.

Das heißt, der Sender darf **maximal** W Datenblöcke senden, ohne auf eine Quittung warten zu müssen (maximal, es gehen auch weniger). W ist also am Sender wie die Zahl der Credits zu interpretieren und am Empfänger als die Größe des Empfangspuffers.

1.3 Aufgabe 3: