Herleitung an den Sliding-Windows

Utopia

Beginnen wir mit den einfachsten und zugleich unwahrscheinlichsten Voraussetzungen, die bei einer Übertragung ideal (und leider eben auch utopisch) sind:

Voraussetzungen:

V1.) Simplex-Übertragung (die Daten werden nur in eine Richtung übertragen)

V2.) Sendende und empfangende (Vermittlungs-)Schicht sind allzeit bereit zum Senden/Empfangen

V3.) Verarbeitungszeiten sind nicht vorhanden

V4.) Puffer ist unendlich groß

V5.) Übertragungskanal zwischen den Sicherungsschichten verliert nie Rahmen und beschädigt sie nie.

Das Einzige, was in diesem völlig unrealistischen Fall passieren würde:
~~Der~~ Die sendende Sicherungsschicht würde sich endlos Pakete von der immer bereiten (V2) Vermittlungsschicht holen und diese völlig fehlerfrei (V5) an die Sicherungsschicht des Empfangsrechners senden. Diese Sicherungsschicht des Empfangsrechners sendet diese Rahmen (die Sicherungsschicht des Senderechners macht aus den Paketen ja Rahmen) stets weiter (ohne Prüfung, denn die Übertragung

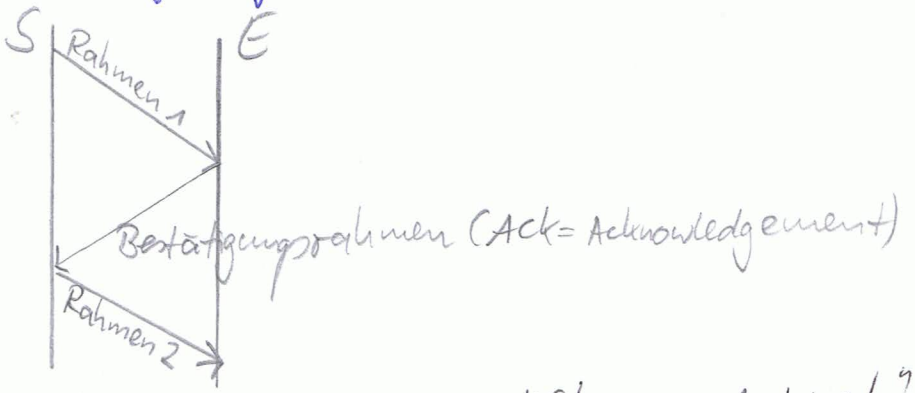
läuft ja fehlerlos (s. V5)) an die immer bereite Vermittlungsschicht des Empfangsrechners.

V2

Doch, wie gesagt, das ist UTOPIE!
Wir legen nach und nach die Voraussetzungen ab und kommen stets zu veränderten Protokollen:

Was nämlich nicht der Fall ist:
Die Empfängerseite kann nicht unendlich viel Information „puffern" (zwischenspeichern, V4).
Weil das so ist, muss darauf geachtet werden, dass der Sender nicht schneller sendet als der Empfänger empfangen kann (Stichwort: Flusskontrolle).

Die einfache Lösung für dieses Problem ist, dass der sendende Rechner erst dann den nächsten Rahmen schickt, wenn er vom ~~Sender~~ Empfänger eine Bestätigung erhalten hat.



Das ist das sogenannte „Stop-and-Wait"-Protokoll.

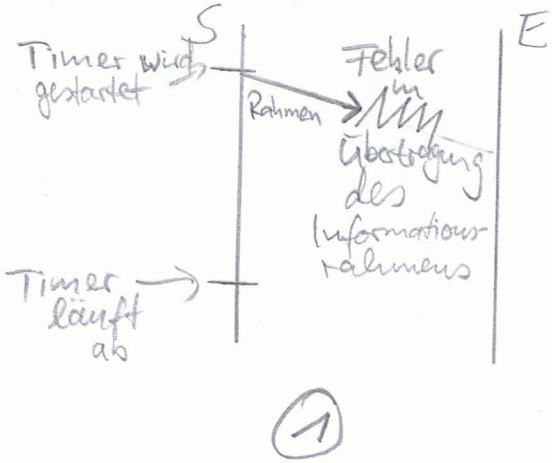
Wir sehen, dass durch das Senden der Empfangsbestätigung (Acknowledgement) der Empfänger zum Sender und der Sender zum Empfänger wird.
Voraussetzung war ja, dass die Rahmen nur in eine Richtung geschickt werden (V1). Nun werden sie zwar immer noch einzeln, aber mittlerweile in beide Richtungen gesendet.
Wir kommen von der Simplex-Übertragung zu einer Duplex-Übertragung (beide Rechner können senden). Allerdings geschieht dies nicht gleichzeitig, sondern stets im Wechsel, deswegen ist es eine Semi-Duplex-Übertragung (auch "Halbduplexübertragung").

Simplex-Protokoll für rauschende Leitungen

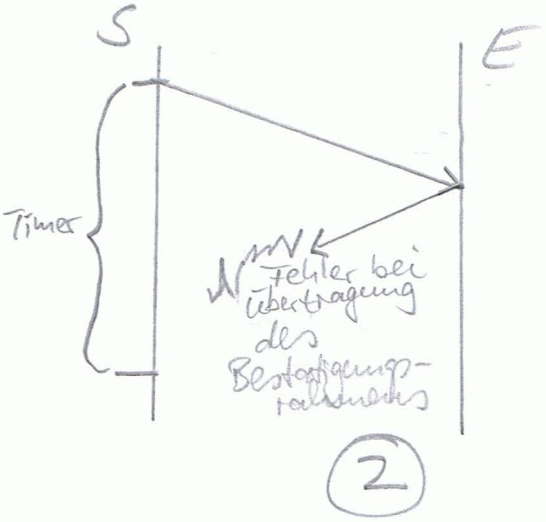
Natürlich ist auch die Voraussetzung, dass der Kanal fehlerfrei ist (V5), utopisch.
Durch die (bereits behandelte) Prüfsumme (Cyclic-Redundancy-Check, CRC) wird beim Empfänger geprüft, ob der gesendete Rahmen Fehler enthält.

Wenn der Rahmen fehlerfrei ist, dann schickt der Empfänger einen (leeren) Bestätigungsrahmen zurück.
Andernfalls (also wenn der Rahmen Fehler hat) wird kein Bestätigungsrahmen geschickt.

Bevor der Sender seinen Rahmen schickt, stellt er eine Uhr, den sogenannten "Timer". Der Timer wird so gestellt, dass er nicht abläuft, bevor der Bestätigungsrahmen (im Falle einer fehlerlosen Übertragung) ankommt. Wenn der Timer abläuft, ohne dass ein Bestätigungsrahmen angekommen ist, hat der Empfänger (scheinbar) keinen Bestätigungsrahmen geschickt und das hieße, die Übertragung war fehlerhaft. ~~Also sendet~~ sendet der Sender den Rahmen erneut.



(1)



(2)

Natürlich kann auch die Variante (2) auftauchen. Der Rahmen wurde fehlerlos übertragen, aber der Bestätigungsrahmen geht verloren. An dieser Stelle würde der Sender den Informationsrahmen ja noch einmal senden. Der Empfänger würde also zweimal den gleichen Rahmen gesendet bekommen - ~~das wäre ja überflüssig~~ im schlimmsten Fall sogar noch öfter...

Der Empfänger muss also bereits angenommene von neuen, noch nicht angenommenen Rahmen unterscheiden können.

Dazu müssen die Rahmen also numeriert werden. Wie sollen sie nun numeriert werden?
Wir sprechen hier von Folgenummern, die den Rahmen mitgegeben werden und im Rahmenheader stehen.
Beim "Stop-and-wait"-Verfahren reicht die wechselnde Numerierung ~~von~~ mit 0 und 1, denn:

→ Bevor Rahmen m+1 gesendet wird, muss Rahmen m bestätigt worden sein.

→ Eine Verwechslung von Rahmen m+2 mit Rahmen m ist ~~nicht~~ somit nicht möglich (denn sowohl m also auch m+2 bekämen beispielsweise die 0 als Folgenummer). Aber m+2 wird ja erst gesendet, wenn m+1 (Folgenummer 1) bestätigt wurde und m+1 wird ja erst gesendet, wenn Rahmen m (Folgenummer 0) bestätigt wurde.

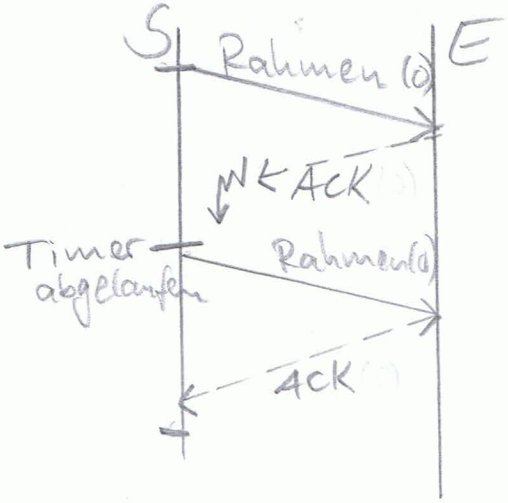
Protokolle, bei denen der Sender auf eine Bestätigung wartet, bevor er die nächsten Daten sendet, nennt man "PAR-Protokolle" (von Positive Acknowledgement with Retransmission) oder auch "ARQ-Protokolle" (von Automatic Repeat reQuest).

Wie gesagt, muss der Timer des Senders lang genug eingestellt sein, damit die Bestätigung noch vor Ablauf des Timers eintrifft.

Der Sender versieht seine Rahmen abwechselnd mit den Folgenummern. Der Empfänger erwartet Rahmen mit bestimmter Folgenummer.

1.) Wenn der Empfänger den Rahmen nicht erhält, weil er verloren gegangen ist (oder er kommt fehlerhaft an), dann schickt er auch keine Bestätigung (logisch). Die Neuübertragung (Retransmission) geschieht, sobald beim Sender der Timer abgelaufen ist!

2.) Bestätigung geht verloren:



Der Sender kann nicht unterscheiden, ob der ~~Rahmen oder die~~ Informationsrahmen oder der Bestätigungsrahmen verloren gegangen ist. Es kann sein, dass die Übertragung des Informationsrahmens fehlerfrei war, aber die Bestätigung verloren geht.
Sobald der Timer abgelaufen ist, überträgt der Sender einfach nochmal

den gleichen Rahmen (hier: Rahmen 0).
Bereits bei der ersten Sendung von Rahmen 0 leitete der Empfänger den Rahmen an die Vermittlungsschicht weiter und erwartet jetzt den Rahmen mit der Sequenznummer 1.
Da erneut die 0 ankommt, weiß der Empfänger, dass im vorigen Durchlauf die Bestätigung verloren gegangen sein muss. Also schickt er einfach nochmal die Bestätigung für Rahmen 0, in der Hoffnung, dass sie diesmal ankommt (der Rahmen 0 wurde ja bereits an die Vermittlungsschicht übergeben, wird nun also verworfen).

Protokolle mit variablen Fenstergrößen

Der "Huckepacktransport" (Piggybacking) (Folie S. 57)

Wir sind ja jetzt im "Halbduplex-Übertragungsverfahren".
Soll heißen: Die Daten werden in beide Richtungen ausgetauscht, jedoch stets nacheinander.
Das "Stop-and-wait"-Verfahren dauert so seine Zeit, denn es darf ja immer erst ein neuer Rahmen gesendet werden, wenn die Bestätigung des Vorrahmens angekommen ist.
Das lässt Überlegungen zu, ob es nicht sinnvoll wäre, für die Bestätigungsrahmen eine Extraleitung zu legen (wir hätten also jetzt einen vollen Duplex-Modus).
Problem: Der Bestätigungsrahmen ist im Vergleich zum Informationsrahmen enorm klein und das Senden der Bestätigung ist Bandbreitenverschwendung.
Der Benutzer müsste für zwei Leitungen bezahlen, wobei eine von beiden fast vollständig ungenutzt bleibt. => nicht gut!

Deswegen kam die Idee des sogenannten Piggybacking's, also des Huckepacktransports. Stellen wir uns eine Brieffreundschaft vor. Egon schreibt Kunigunde einen Brief. Kunigunde erhält den Brief und müsste, nach bisher bekannten Verfahren, eine Bestätigung an Egon schicken. Dieses Verfahren können wir eventuell