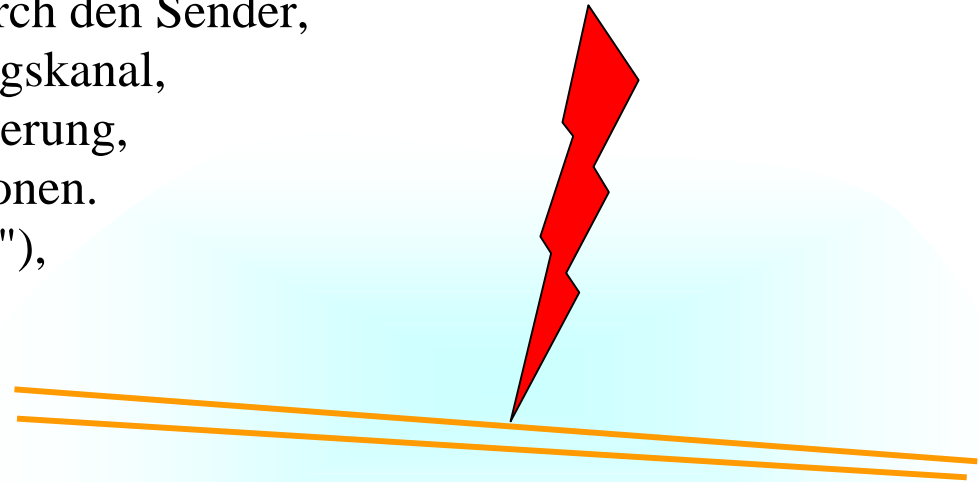


6. Grundlegende Protokollmechanismen

6.1. Prüfsummen/CRC Fehlersicherung

- Ursachen von Übertragungsfehlern:
 - Absichtlicher Abbruch der Meldung durch den Sender,
 - physikalische Störungen im Übertragungskanal,
 - ausser Tritt geraten der Taktsynchronisierung,
 - HW- & SW-Fehler in den Partner-Stationen.
 - Überlastung des Empfängers ("Overrun"),
 - Überlastung des Senders ("Underrun"),
 - Zugriffskollisionen im LAN,
 - Fehler in Drittstationen.
- Fehlercharakteristiken:
 - Fehlerwahrscheinlichkeit (10^{-2} .. 10^{-15}),
 - zufällige und periodische Fehler,
 - Bitfehler oder Burstfehler.
- Diese Störungen sollen erkannt und falls möglich korrigiert werden:
 - Byteparität, Langs- und Querparität,
 - Zyklische Redundanzprüfung "CRC",
 - fehlerkorrigierende Codes.



6.1.1 Paritätsprüfungen

- z.B. 8-Bit Code wird um Paritätsbit erweitert. -> Start-Stop DFÜ
- Paritätsoptionen:
 - insbesondere bei Start-Stop Betrieb,
 - Even / Odd / None / Zero / One.
- Vorwärts-Fehlerkorrektur, "Forward Error-Correction" ohne Rückfrage.
- Mit Längs- & Querparität kann man einzelne Fehler sogar korrigieren:

STX	M	E	L	D	U	N	G	.	7	ETX	CHK
0	1	1	0	0	1	0	1	0	1	1	0
1	0	0	0	0	0	1	1	1	1	1	1!
0	1	1	1	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	1	0	0
0	1	1	1	1	1	1	1	0	0	0	1
1	0	1	1	0	0	1!	0	0	1	0	1

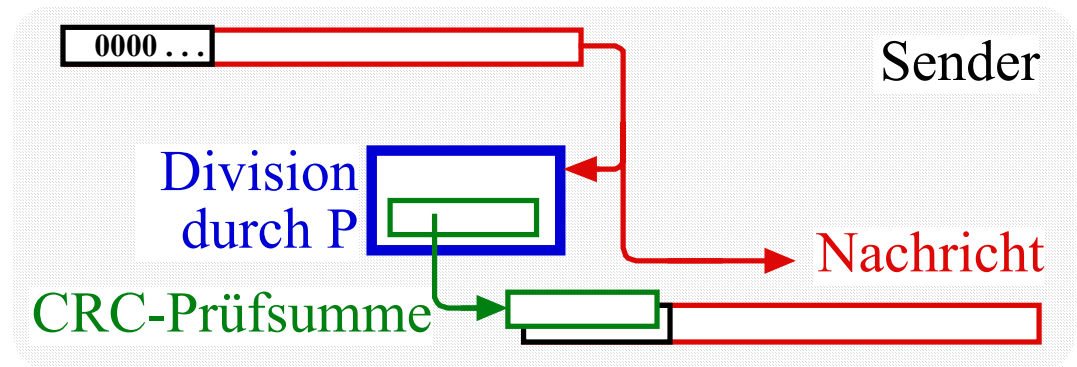
6.1.2 Zyklische Prüfsummen ("CRC")

- Wesentlich verbesserte Fehlererkennung im Vergleich zur einfacher Summenbildung.
- Die Länge L der Prüfsumme variiert zwischen 12, 16 oder 32 Bit.
- Wird ein Fehler erkannt, verwirft der Empfänger die Nachricht.
- Sender wiederholt dann die Nachricht.
- Übertragungsfehler wird erkannt:
 - wenn Fehlersequenz kürzer als 16 bzw. 32 Bit,
 - wenn Anzahl der Fehlerbits 1,2 oder ungerade,
 - und 99,99% aller längeren Burstfehler.
- Modulo 2 Arithmetik:
 - $0+1 = 1$ $1+0 = 1$ $0+0 = 0$ $1+1 = 0$
 - $0-1 = 1$ $1-0 = 1$ $0-0 = 0$ $1-1 = 0$
 - Multiplikation als sukzessive Addition,
 - Division als sukzessive Subtraktion,
- XOR-Funktion leicht in Hardware implementierbar (XOR-Gatter).

=> XOR-Funktion: $a \oplus b$

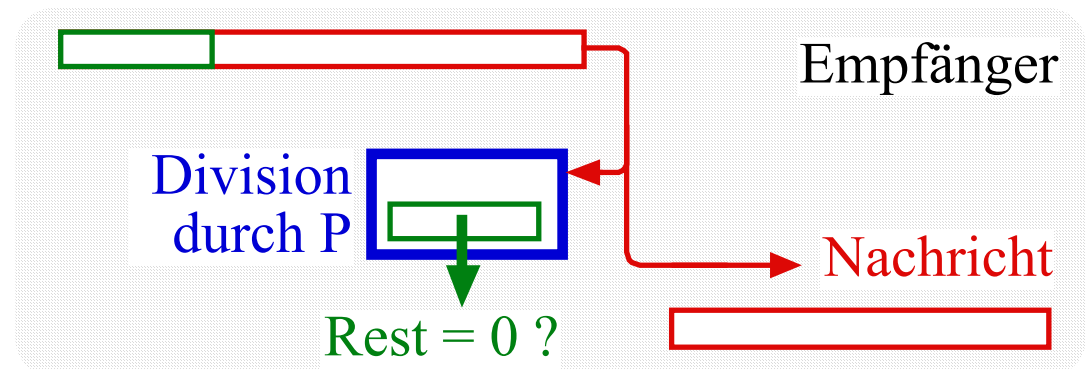
- Erzeugen der CRC-Prüfsumme:

- Nachricht mit 2^L multiplizieren,
- Nachricht durch eine feste Prüfwahl P dividieren,
- Division geschieht ohne Überträge (Modulo 2),
- Divisionsrest von Nachricht subtrahieren,
- Paket zum Empfänger übertragen:



- Validieren der Prüfsumme beim Empfänger:

- Paket durch feste Prüfwahl P teilen,
- falls Divisionsrest = 0, dann Nachricht OK,
- Paket aufteilen in Nachricht und Prüfsumme,
- Nachricht abliefern (falls OK) .



- **Alternative Interpretation:**

- ein b Bit langer Bitstrom wird als Polynom des Grades $b-1$ aufgefaßt,
- z.B. werde $89 = 1000\ 1001$ interpretiert als x^7+x^3+1

- **Geeignete Prüfpolynome sind etwa:**

a) CRC-12: $x^{12} + x^{11} + x^3 + x^2 + x + 1$

b) CRC-16: $x^{16} + x^{12} + x^2 + 1$

c) CRC-V.41: $x^{16} + x^{12} + x^5 + 1$

d) CRC-32: $x^{32} + x^{26} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^4 + x^2 + x + 1$

- **bzw. binär:**

- 1 1000 0000 1111,
- 1 0001 0000 0000 0011,
- 1 0001 0000 0010 0001,
- 1 0000 0100 0100 0001 0001 1101 1001 0111 .

6.1.3 Fehlererkennungsleistung von CRC Prüfsummen:

- Verfälschte Meldungen durch Addition eines Fehlerpolynoms $E(x)$.
- Der Fehler wird entdeckt, wenn auch das Fehlerpolynom $E(x)$ **nicht** durch das Prüfpolynom $P(x)$ teilbar ist:

$$E(x) \cdot x^m / P(x) \neq 0$$

- Entdeckt werden folgende Fehler:
 - kurzer Burst: $E(x)$ ist sicher nicht teilbar durch $P(x)$, falls $E(x)$ kürzer als $P(x)$.
 - 1 Bit Fehler: $E(x) = x^i$ ist nicht teilbar durch $P(x)$, da $P(x)$ Primfaktor x nicht hat..
 - Doppelfehler: $E(x)$ lässt sich dann darstellen als $x^i(x^{j-i}+1)$. P'faktor x in $P(x)$ fehlt.
 - odd Bitfehler: $E(x) \neq (x+1) Q(x)$, wird entdeckt falls $P(x)$ als $P(x) = (x+1) R(x)$ gewählt wird. Jedes Polynom mit $(x+1)$ als Faktor hat jedoch eine gerade Anzahl Terme. $E(x)$ kann also $(x+1)$ nicht als Faktor enthalten.

⇒ Peterson, W. & Brown, D. "Cyclic Codes for Error Detection", Proc. IRE, January 1961.

⇒ Jonathan Cook, New Mexico State University,
www.cs.nmsu.edu/~jcook/Classes/DE-CS484/DataLink-3.html,

6.1.5 Realisierung durch Software:

- Nachbildung des Divisionsalgorithmus.

```

const  crcPoly = $13; frontbit = $10;
var    register, xorInput : longint;

procedure AddBit( MessageBit : integer);
begin  if (register >= frontbit)
        then xorInput := crcPoly
        else xorInput := 0;
        register := register XOR xorInput;
        register := register SHL 1;
        register := register + MessageBit;
        (* Now print register value *)
end;

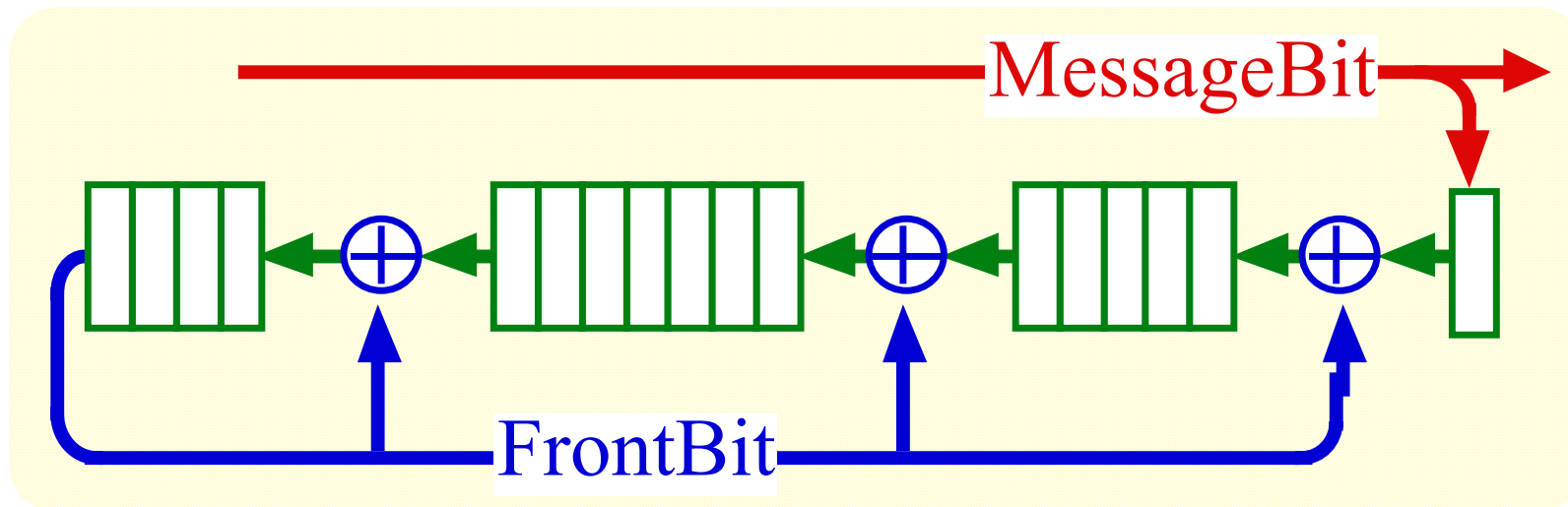
begin .... (* add all MessageBits *) end.

```

register:	MessageBit
0000 1	1
000 11	1
001 10	0
011 01	1
110 10	0
10011	1
000 01	1
000 10	0
001 01	1
010 10	0
101 00	0
011 10	0
111 00	0
111 10	0
1101	

6.1.6 CRC-V.41 mit Schieberegister:

- Rückkopplung über 16 Stufen (CRC-V.41).
- Entsprechend Division und Softwarelösung.
- Registerinhalt pro Takt 1 Stelle verschoben.
- Einspeisung ins Schieberegister für Polynomkoeffizienten $\neq 0$.
- Der im Schieberegister verbliebene Rest wird anschliessend an das letzte Bit der Meldung übertragen.
- Bitstuffing geschieht nach der CRC-Bildung.



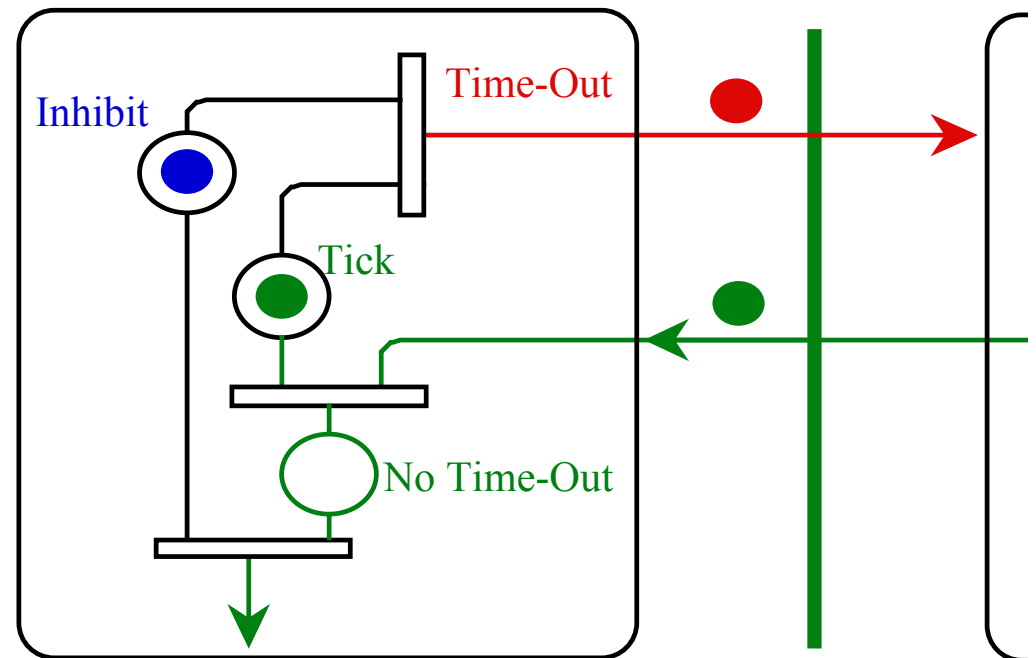
6.1.7 Beispiel für Vorwärts-Fehlerkorrektur

- Redundanzpakete hinzufügen.
- Verlorene Pakete rekonstruieren.
- z.B. 3 Pakete sind zu übertragen:
P1: 10101010
P2: 00110011
P3: 00001111
- Redundantes Paket P4:
P4: 10010110 ergibt sich aus $(p1 \text{ xor } p2: 10011001) \text{ xor } P3$
- Rekonstruktion des verlorenen P2, z.B.:
 $P2 = P4 \text{ xor } (P1 \text{ xor } P3)$
 $P2 = P4 \text{ xor } (10100101)$
 $P2 = 00110011$ (sic)
- Entsprechend lässt sich P1 oder P3 rekonstruieren.
- Es muss bekannt sein, welches Paket fehlt.

6.2. Bestätigungen

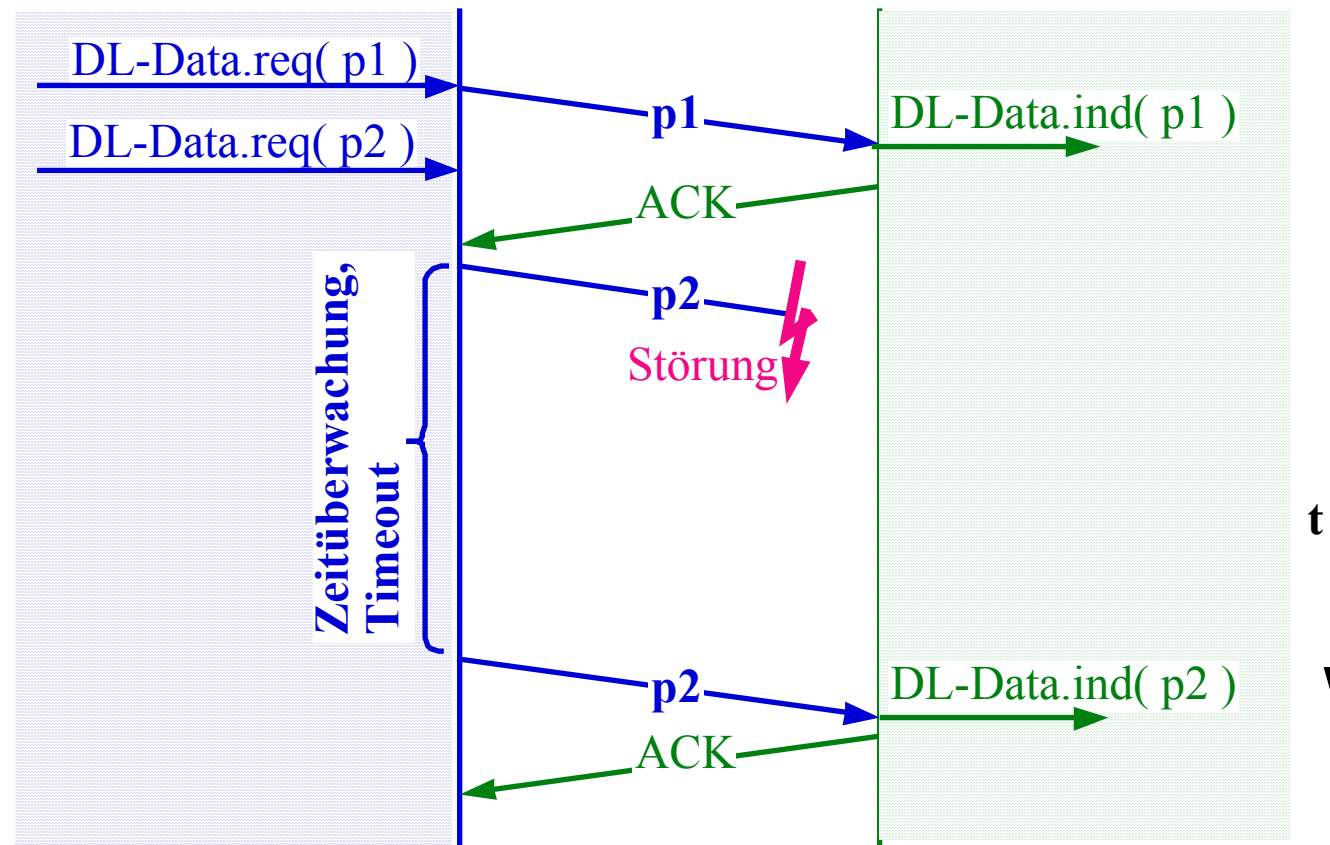
6.2.1 Zeitüberwachung ("Time-Out")

- Die Aktionen eines Senders oder Empfängers werden vom Ablauf eines lokalen Zeitgebers abhängig gemacht:
 - Nachricht wiederholen, wenn die Bestätigung ausbleibt,
 - "Lebenszeichen" zum Kommunikationspartner,
 - zeitlichen Abstand zw. Nachrichten einhalten,
 - Feststellen eines Leitungsunterbruchs,
 - Abbau der Verbindung.
- Petrinetz-Darstellung:
 - **Tick** erzeugt neue Marken im Zeitabstand T nach einer Entladung,
 - **Inhibit** erzeugt Marken im Abstand $T+\varepsilon$ und verhindert nicht-deterministische Zündung.



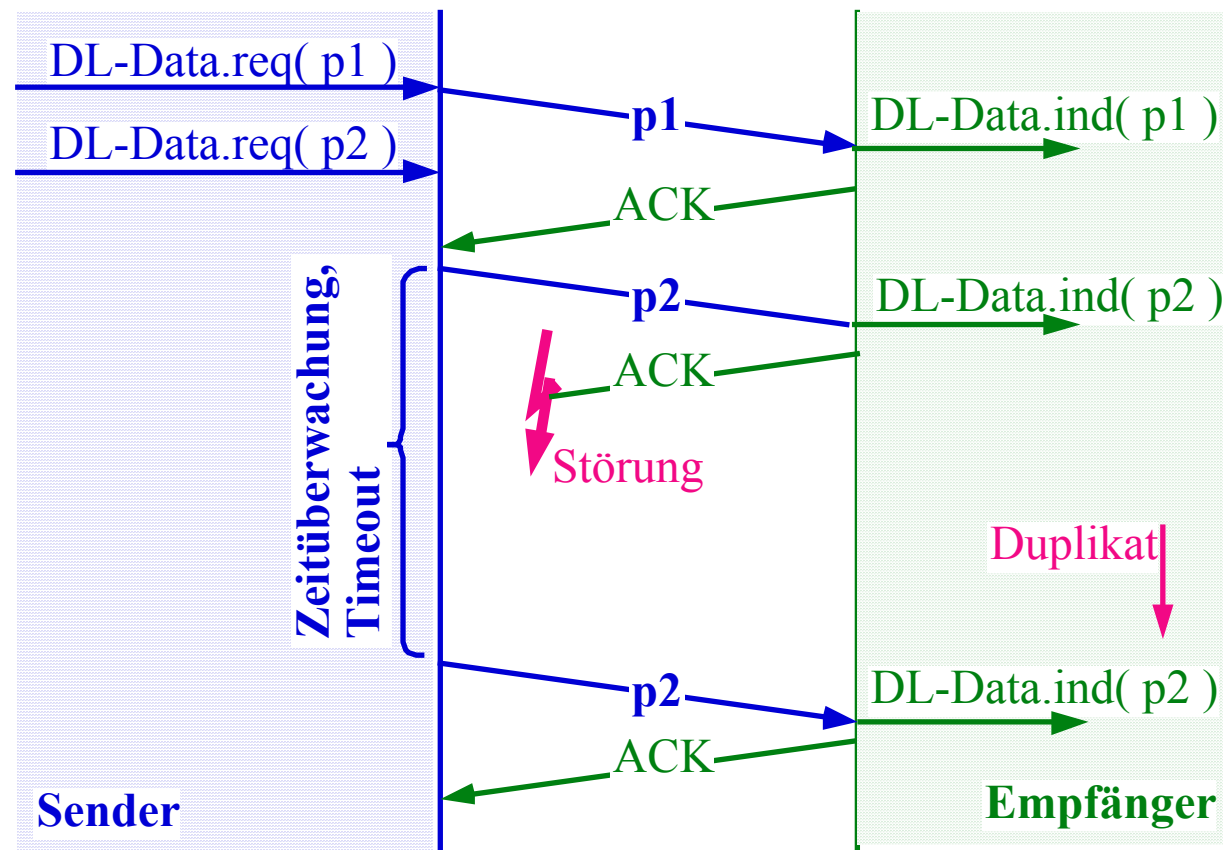
6.2.2 "Stop & Wait" Bestätigung

- Nachricht übertragen & dann auf Quittung warten (ACKnowledge).
- Wenn Quittung ausbleibt, wird die Übertragung wiederholt.
- Die Wiederholung erfolgt erst nach Timeout.



Aber:

- Risiko besteht, dass Meldung doppelt übertragen wird - und zwar dann, wenn die Quittung verloren geht:



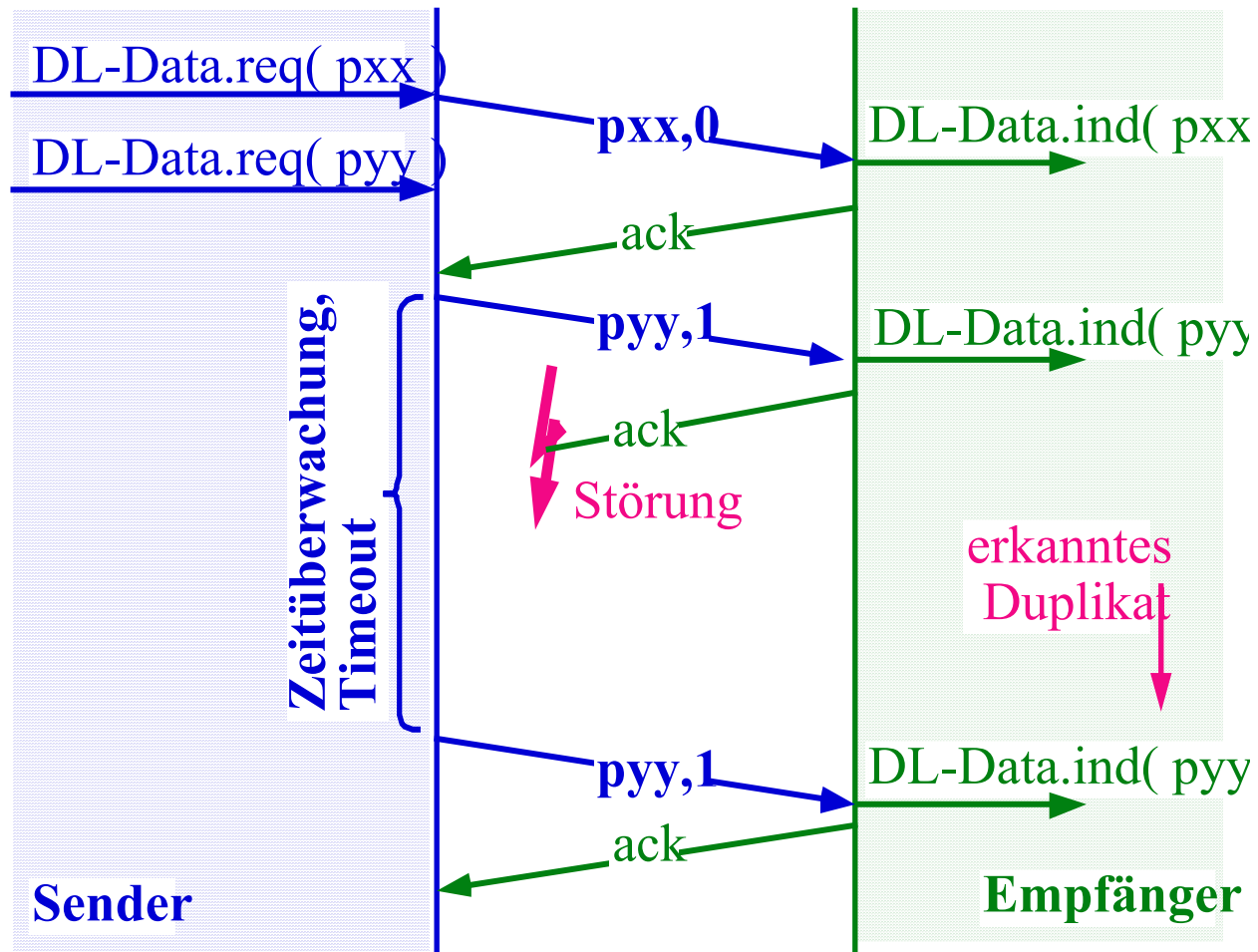
=> Lösung hierzu: **Pakete müssen nummeriert werden.**

6.2.3 0/1 Nummerierung der Pakete

- Der Empfänger erwartet abwechselnd Pakete mit Nummer #0 und #1.
- Nummern 0 & 1 ausreichend, da immer nur eine unbestätigte Nachricht.
- Nummernbereich umfasst 2 Werte, „Übertragungsfenster“ maximal 1.
- Wiederholung bei zerstörter Quittung oder zerstörter Nachricht:
- Duplikat wird vom Empfänger als solches identifiziert.
- Teilpakete mit falschem CRC verwerfen.
- Nachteil von „Stop & Wait“ – mit oder ohne Seq#:
 - Übertragungskapazität geht verloren, wenn die Quittungslaufzeit grösser wird,
 - in der Quittungsphase läuft die Leitung leer,
 - insbesondere bei vielen Zwischenknoten.

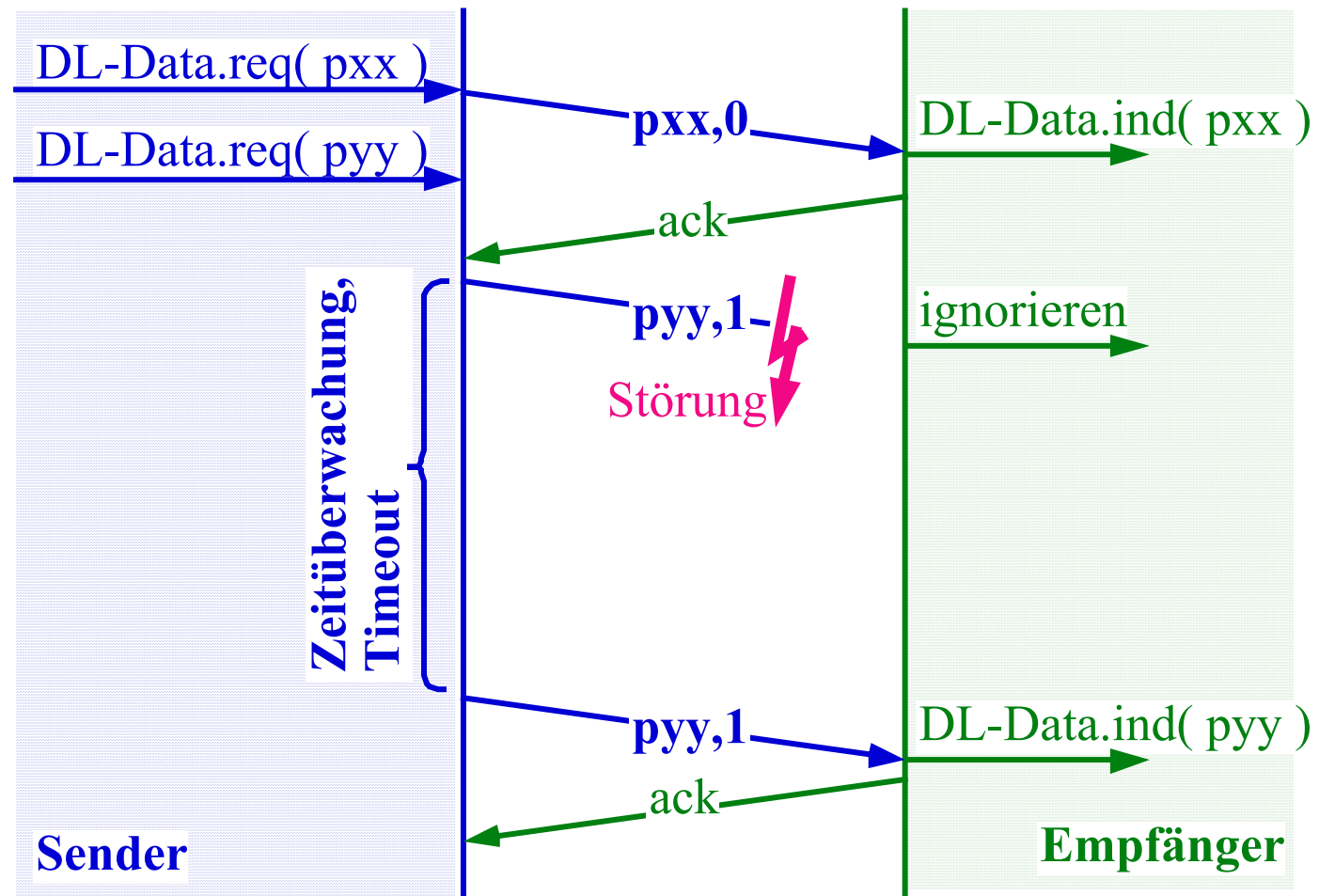
- **Verlorene Quittung:**

- Quittung wird wegen falschem CRC weggeworfen, oder sie geht ganz verloren,
- Erneute Übertragung der Nachricht nach Timeout beim Sender,
- Hier kein Timeout beim Empfänger eingerichtet,
- Sender macht Time-Out.



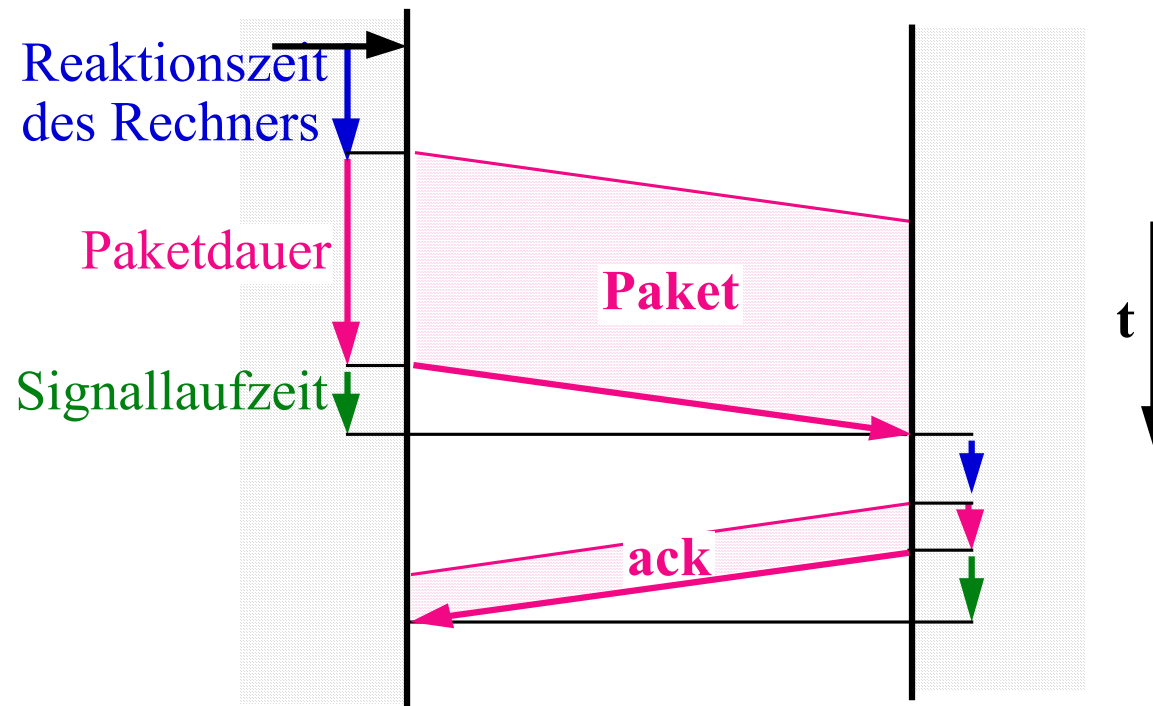
- Verlorene Nachricht:

- Nachricht wird weggeworfen, wegen falschem CRC oder geht ganz verloren,
- Erneute Übertragung nach Timeout beim Sender.



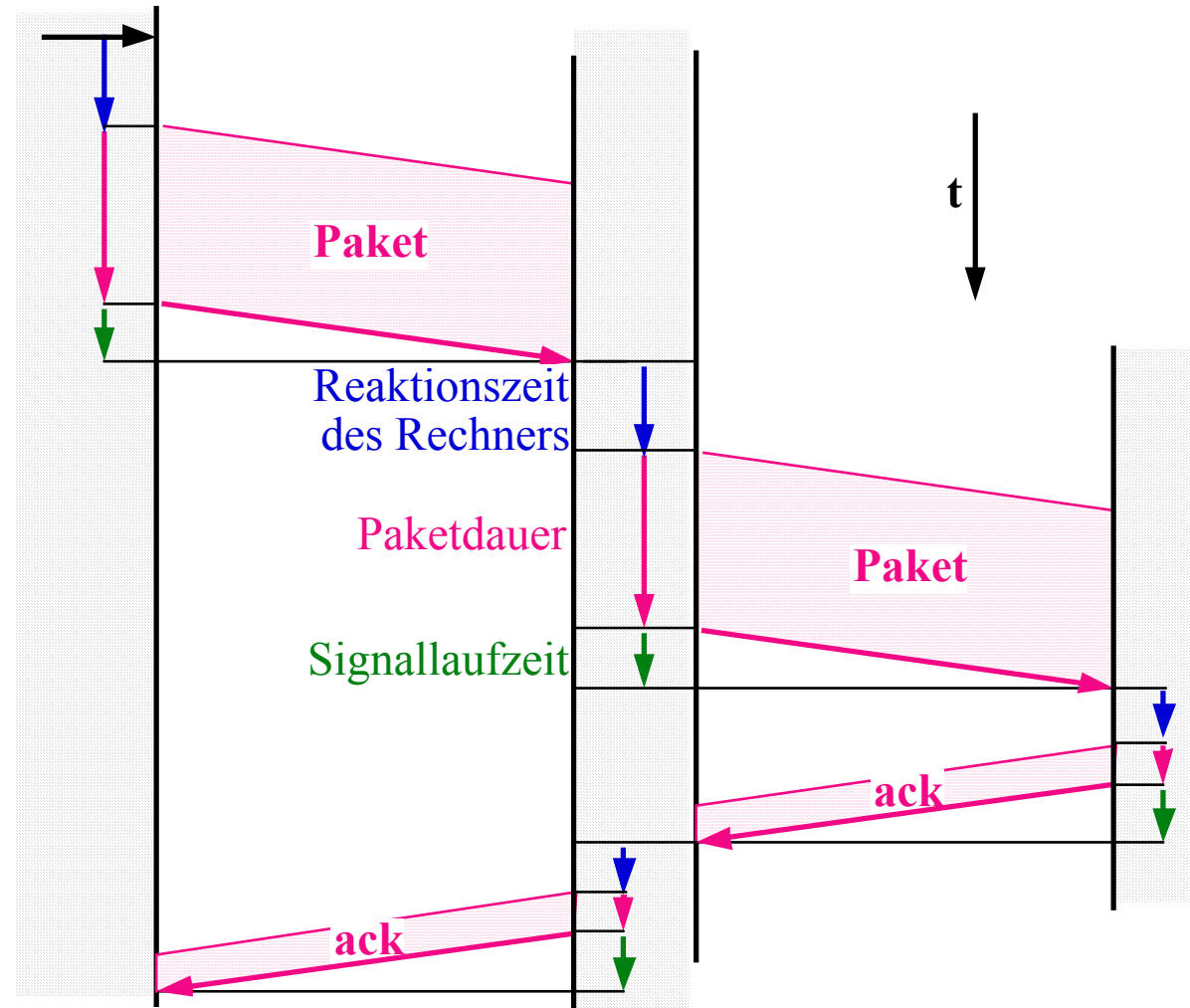
6.2.4 Laufzeiten von Nachrichten:

- Genauere Darstellung der Zeitverhältnisse beim quittierter DFÜ:
 - Direkte Leitung zwischen zwei Knoten:
 - Oft dominiert die Verzögerung im Rechner.
 - Kurze Nachrichten & Bestätigungen belegen die Leitung weniger lang.



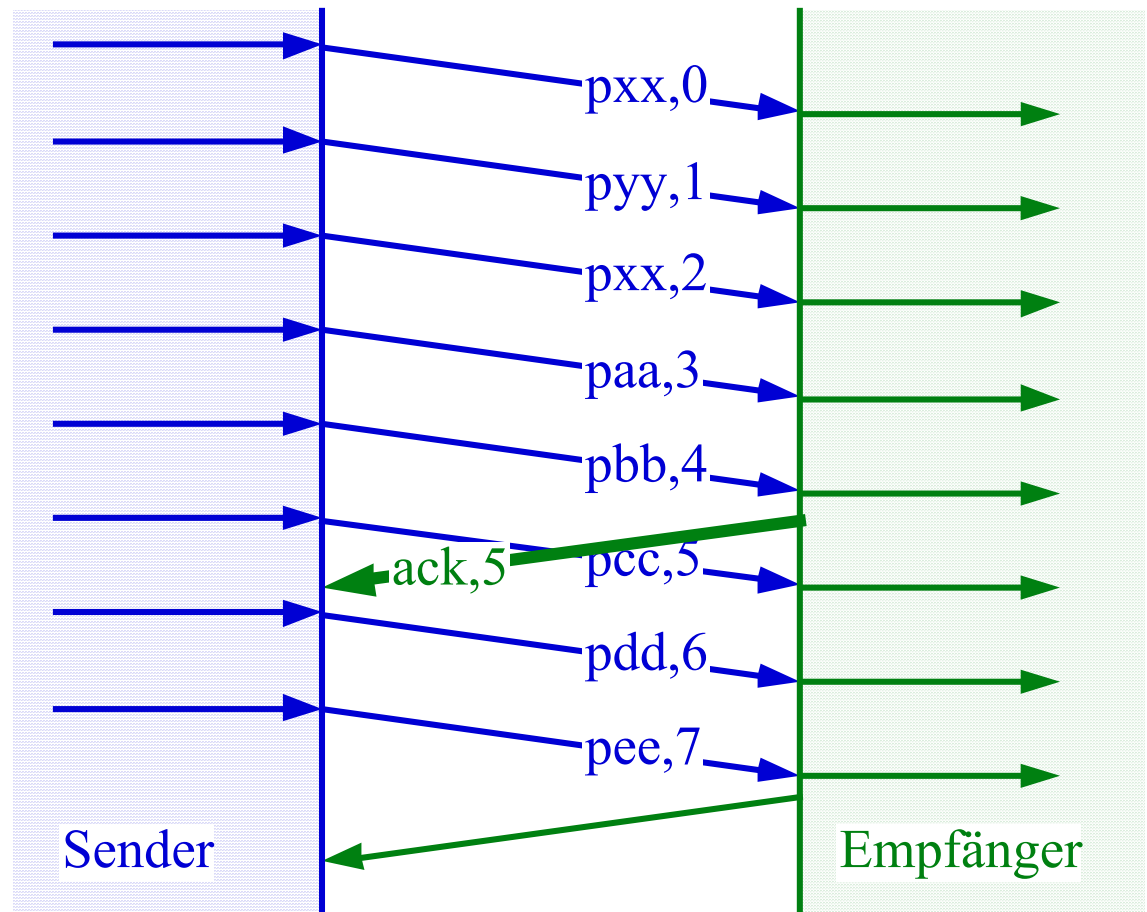
- „Store-&-Forward Delay“:

- Zusätzliche Verzögerung durch die "Store-and-Forward"-Funktion im Zwischenknoten,
- "Cut-through"-Routing im Zwischenknoten ist nicht üblich.
- Die Prüfsumme wird normalerweise abgewartet.



6.3. Fenstermechanismen

- Zu einem Zeitpunkt sind noch mehrere Nachrichten unbestätigt.
- Grösserer Bereich für die Sequenznummern der Nachrichten:
- Meist nächste erwartete Nummer in Bestätigung genannt (z.B. **ack,5**).
- Mehrere Nachrichten mit einer Antwort bestätigen.
- Bessere Leitungsauslastung.
- Nicht nach jeder Nachricht warten.

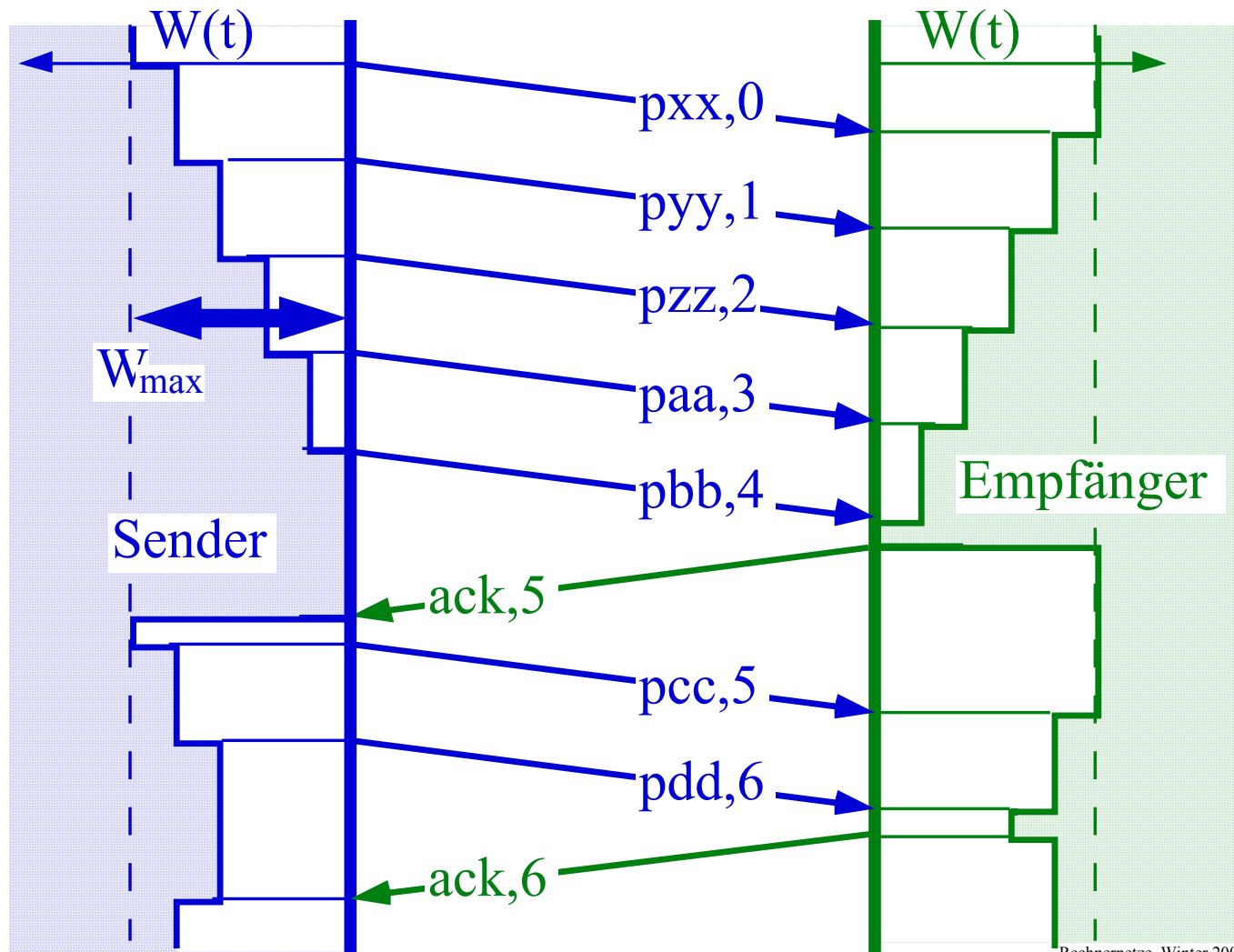


6.3.1 Fensteröffnung

- Maximale Fensteröffnung W_{\max} Pakete:
 - mindestens um 1 kleiner als Nummernbereich,
 - meist als feste Grösse vereinbart.
- Aktuelle Fensteröffnung W (=Window):
 - Fenster W variiert während der Übertragung,
 - Empfänger hält zumindest W Paketpuffer bereit,
 - Sender bewahrt unbestätigte Pakete auf,
 - Sender stoppt nach W unbestätigten Paketen.
- Die Puffer für bestätigte Pakete werden vom Sender freigegeben.
- Empfänger bestätigt Paket P , wenn er genügend Puffer für die Pakete $P+1$ bis $P+W$ hat.
- Wrap-Around der Sequenznummern.
- Nummernbereich $0..n$ erlaubt $n-1$ unbestätigte Nachrichten.
- Grosse Fensteröffnungen z.B. für Satellitenstrecken.

Ablauf der Fensteröffnung:

- Unterschiedliche Sicht beim Sender und beim Empfänger.
- Sender stoppt bei ausgeschöpftem Fenster (=Flusskontrolle).
- Wiederholung falls länger keine Bestätigung.

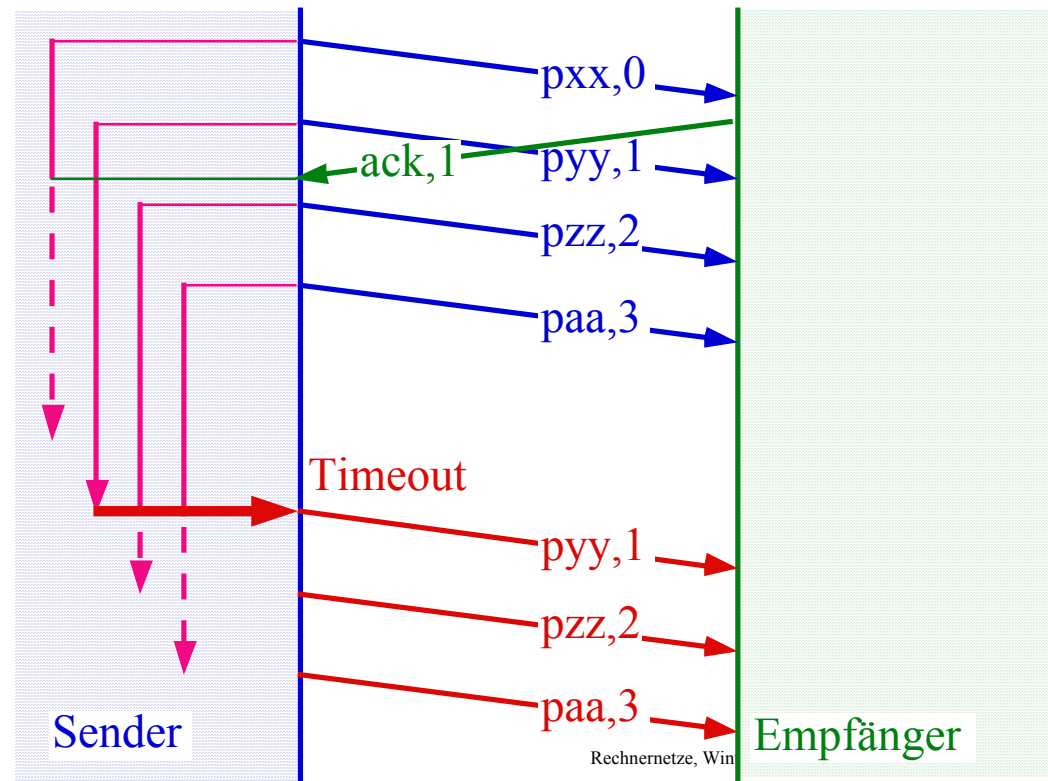


6.3.2 Fehlerbehandlung

- Timeout muss so gross eingestellt werden, daß normalerweise vorher eine Antwort eintrifft.
- Trifft eine Quittung vor Ablauf des Timer ein, so wird einfach das Fenster wieder geöffnet.
- Am besten für jedes Paket ein Timer.

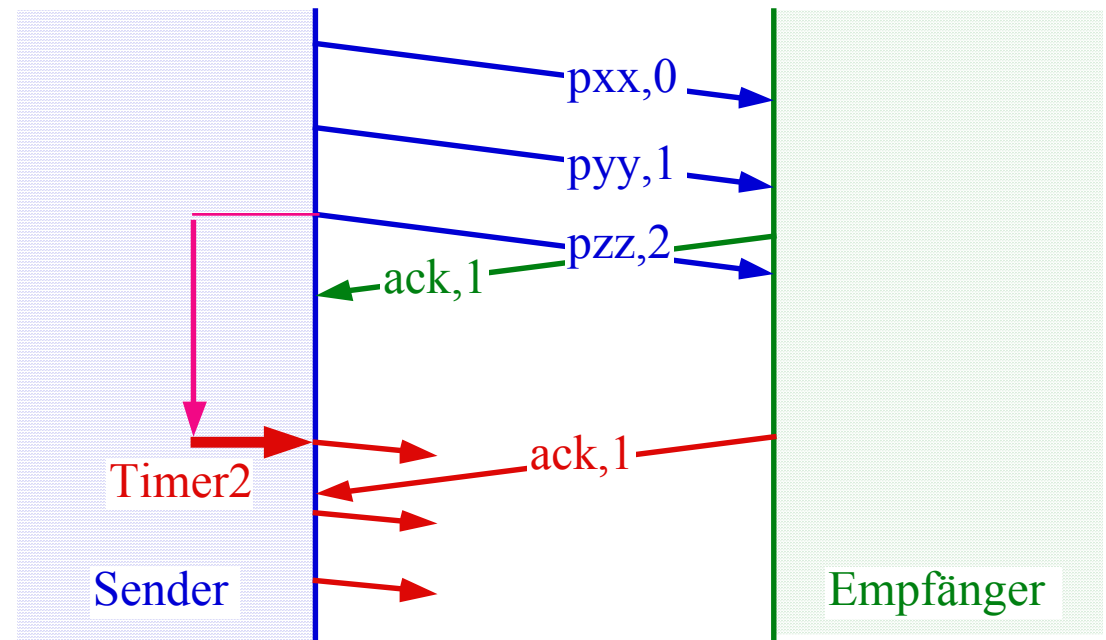
Implizierte Paketwiederholung Fall A:

- Läuft der Timer ab, so werden alle unbestätigten Pakete wiederholt.
- Timer für das Paket (p_{yy,1}) läuft ab.
- Pakete ab (p_{yy,1}) werden wiederholt.



Implizierte Paketwiederholung Fall B:

- Eventuell bestimmt ein zweiter Timer, ab welchem Zeitpunkt eine Quittung gleichzeitig als "Reject" interpretiert wird.
- Wird "nach" dem Senden eines Paketes eine Quittung für ein altes Paket erhalten, so werden die noch unbestätigten Pakete wiederholt.



Negative Bestätigung als "Go-back-to-N":

- Besondere Kontrollnachricht (REJ) fordert Wiederholung ab Paket N.

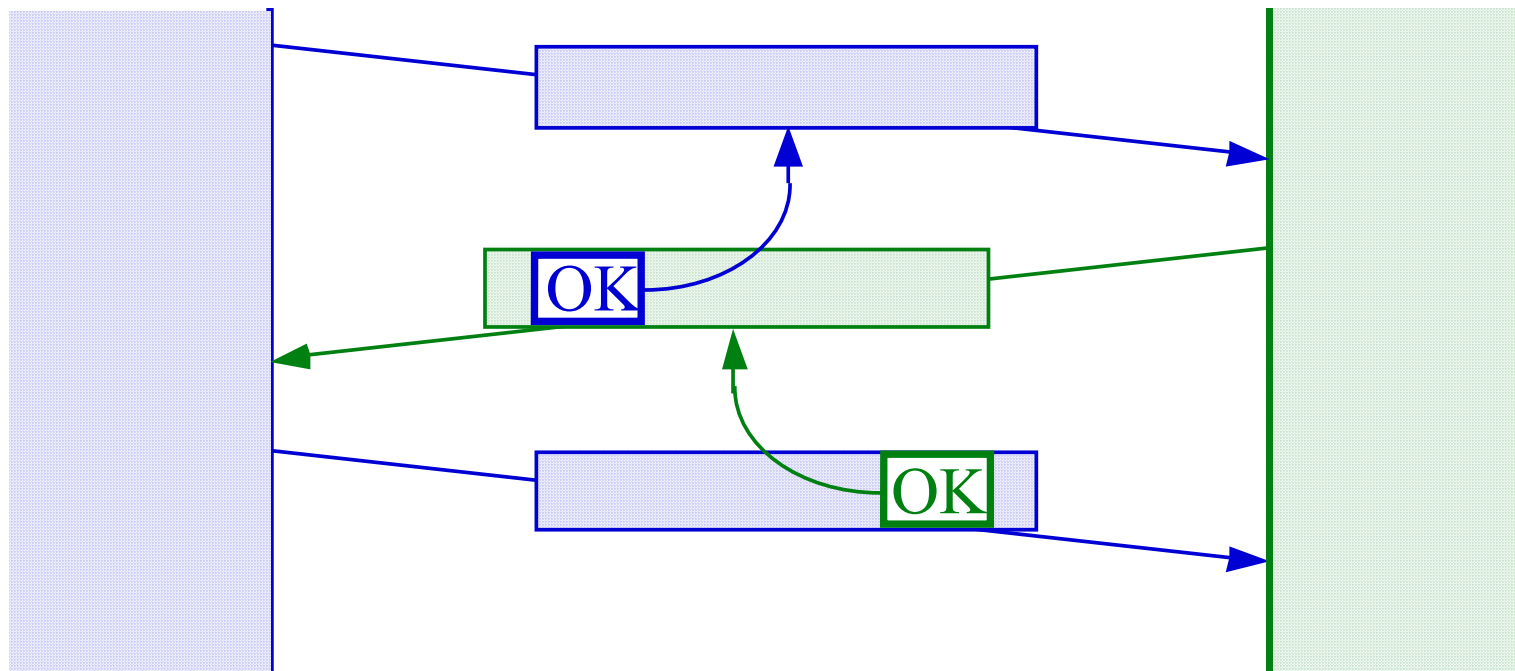
Selektives Reject - SREJ:

- Nur das verlangte Paket wird wiederholt.

6.4. Huckepack-Transport

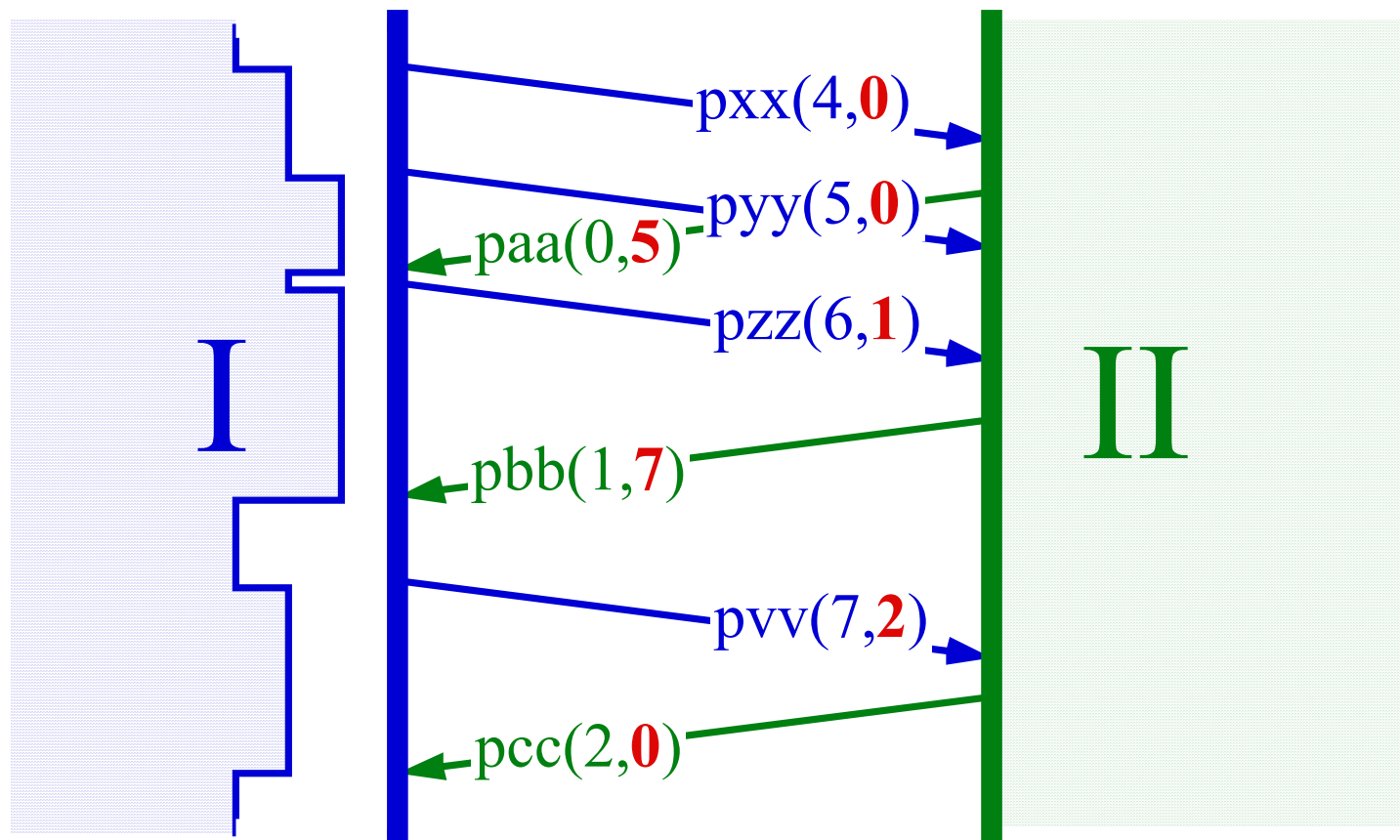
6.4.1 Fenstergröße 1:

- Englisch "Piggy Back"-Transport.
- Bestätigung in den Kopf einer Nachricht in Gegenrichtung verpackt:
- Wenn sowieso Nachricht in Gegenrichtung ansteht, so sind Kosten für Bestätigung klein. Anderenfalls separate Kontrollnachricht.



6.4.2 Transport in beide Richtungen:

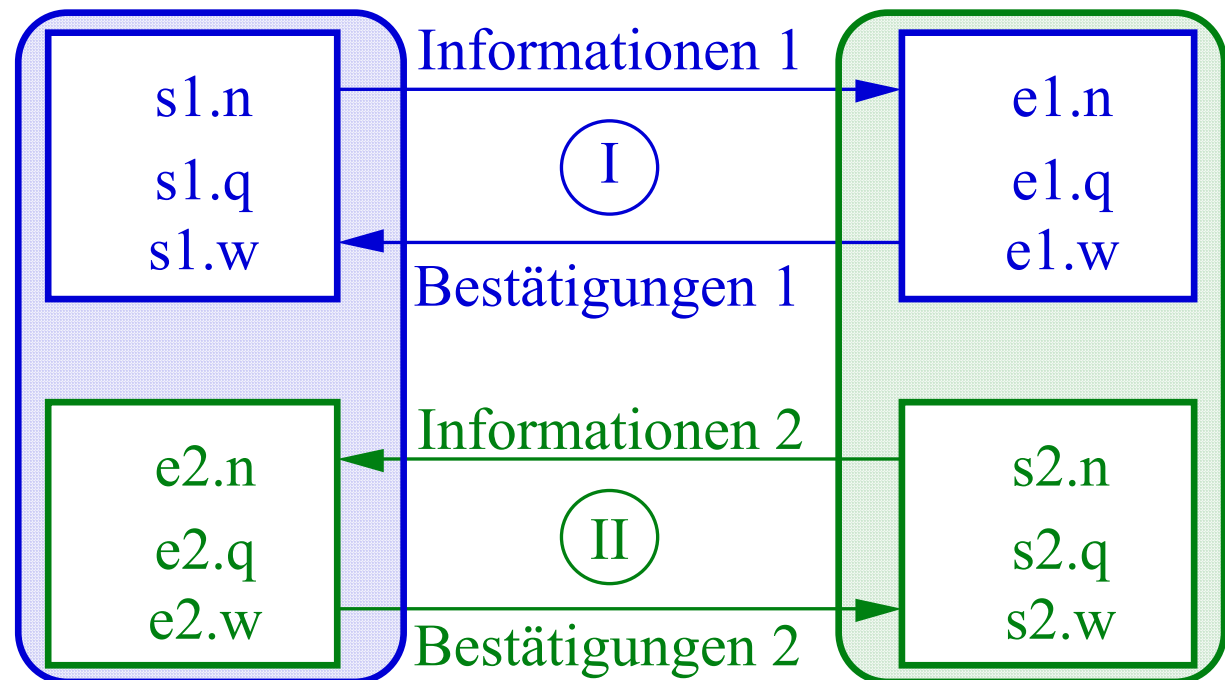
- Huckepack-Quittungen **rot** bzw. **fett** markiert.
- Separate Nachrichtennummern in beide Richtungen z.B. jeweils 0..7 .
- Nachrichten 0..3 zur Station II seien schon übertragen und bestätigt.



6.5. Zustandsautomaten

- Protokolle werden meist mithilfe von Zustandsvariablen implementiert.
- In jedem Knoten läuft ein endlicher Automat.
- Das Ziel ist eine Übereinstimmung der Zustände in den kommunizierenden Knoten.
- Legende:

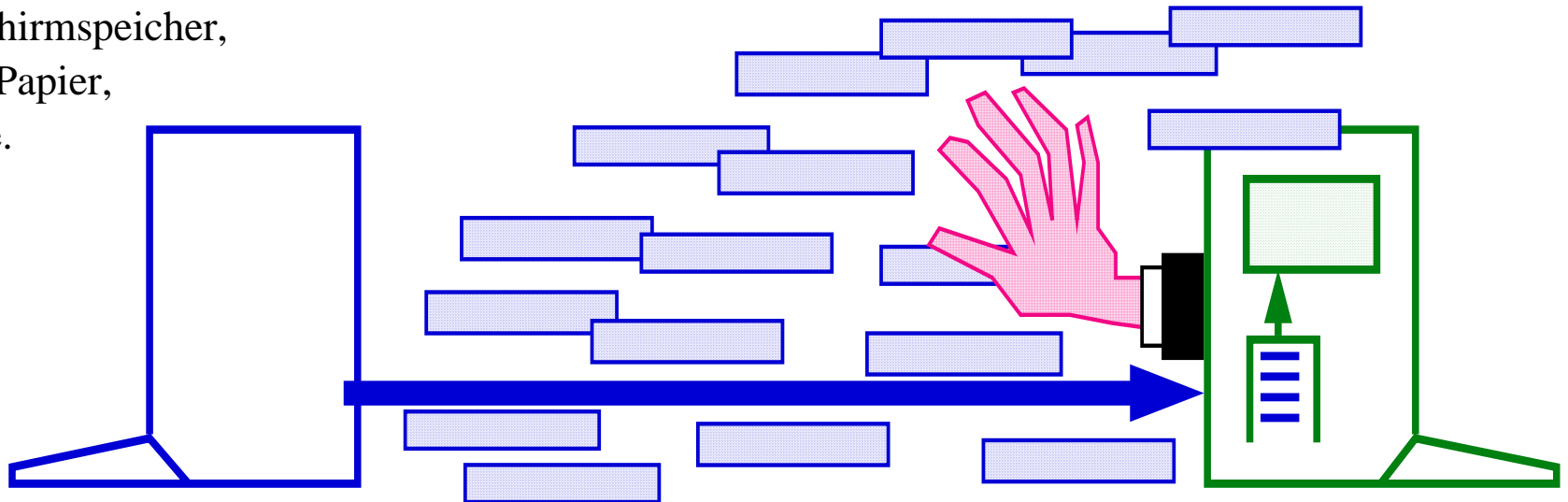
N - Nachrichtensequenznummer
q - Quittungsnummer
w - Fensteröffnung
e1, e2 - Empfänger
s1, s2 - Sender



6.6. Flusskontrolle

6.6.1 Weshalb Flusskontrolle?

- Schutz eines Empfängers vor einer Überflutung durch Datenpakete:
- Suspendieren des Datenflusses wegen Überlastung im Empfänger.
- Anwenderprogramm rechnet zu langsam.
- Keine Puffer mehr für Meldungen.
- Empfänger schreibt nicht schnell genug:
 - auf die weiterführende Datenleitung,
 - in den Bildschirmspeicher,
 - Drucken auf Papier,
 - auf Festplatte.

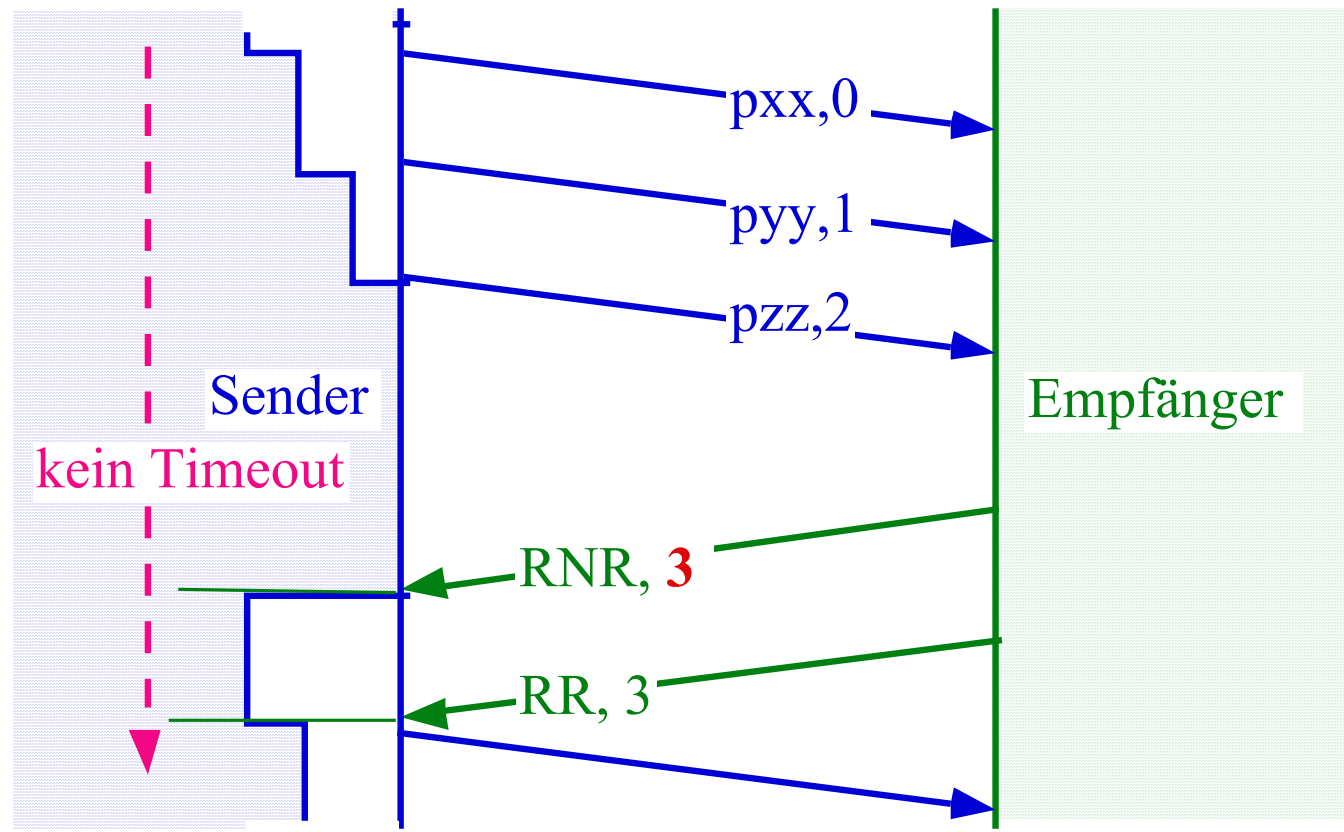


6.6.2 Explizite Flusskontrolle:

- Separate Signale werden vom Empfänger geschickt.
- Für asynchrone Datenströme oft mit Sonderzeichen:
 - Datenstrom anhalten: cntl-S, x-off, DC3,
 - Datenstrom weiter: cntl-Q, x-on, DC1,
 - N.B. das cntl-Präfix erzeugt Zeichen <32.
- Hardware-Flusskontrolle an der seriellen Schnittstelle mit:
 - RTS: Request to Send (Signal zum Modem),
 - CTS: Clear to Send (Signal vom Modem).
- Handshake-Protokoll/Signal an der parallelen Druckerschnittstelle.
- Bei paketorientierten Protokollen oft besondere Kontrollnachricht:
 - RR: Receive Ready
 - RNR: Receive not Ready.

6.6.3 Fensterbasierte Flusskontrolle:

- Ist Fenster ausgeschöpft und hält Sender Bestätigung zurück, so entsteht Flusskontrollwirkung.
- Bestätigung nicht zu lange zurückhalten, sonst geschieht ein Timeout und eine erneute Übertragung.
- Vor Ablauf des Timers bestätigen & expliziten Flusskontrollbefehl schicken:

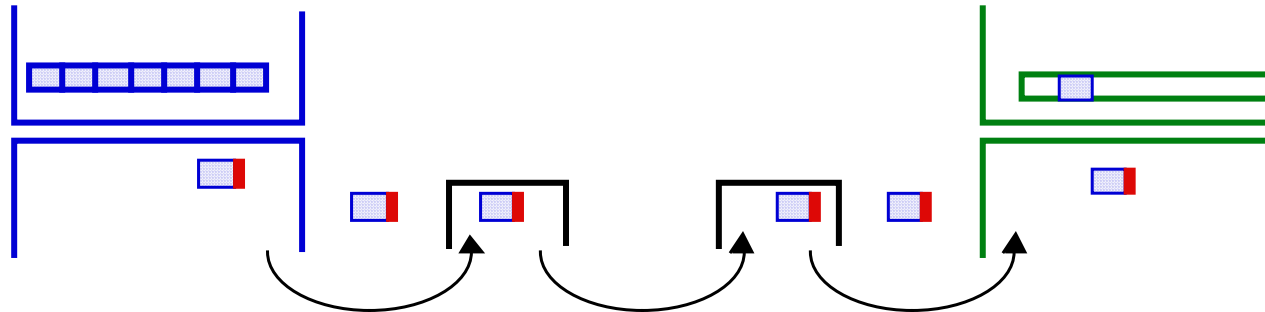


6.6.4 Übertragungsratensteuerung:

- Englisch: "Rate Control".
- Der Sender ist dafür verantwortlich, die vereinbarte Übertragungsrate einzuhalten:
 - mittlere Übertragungsrate,
 - Dauer der Spitzenlast,
 - Spitzenbelastung,
 - garantierte Rate.
- Für Transportsysteme mit langen Bestätigungszeiten. Dann kommt die explizite Flusskontrolle zu spät.
- Für Transportsysteme mit sehr vielen Nachrichten im Transit.
- Wird die Bandbreiten-Allozierung überzogen, so darf das Netz Datenpakete verwerfen.
- Bestandteil der ATM-Dienstgüeverhandlung (QoS) ist Verhandlung der zulässigen Datenrate.

6.7. Paketisierung / Segmentierung

- = Aufspaltung einer längeren Nachricht in kleinere Pakete bzw. Segmente.



- Vorteile:
 - reduzierter Pufferbedarf in den Zwischenknoten,
 - weniger Paketverluste bei schlechten Leitungen,
 - reduzierter "Store & Forward"-Delay.
- Nachteile:
 - zusätzlicher Header für die Pakete,
 - Paketreihenfolge muss sichergestellt werden,
 - erhöhter Kopieraufwand beim Empfang,
 - Reassembly Puffer beim Empfänger.
- Keine Monopolisierung der Leitung durch einen Nutzer.

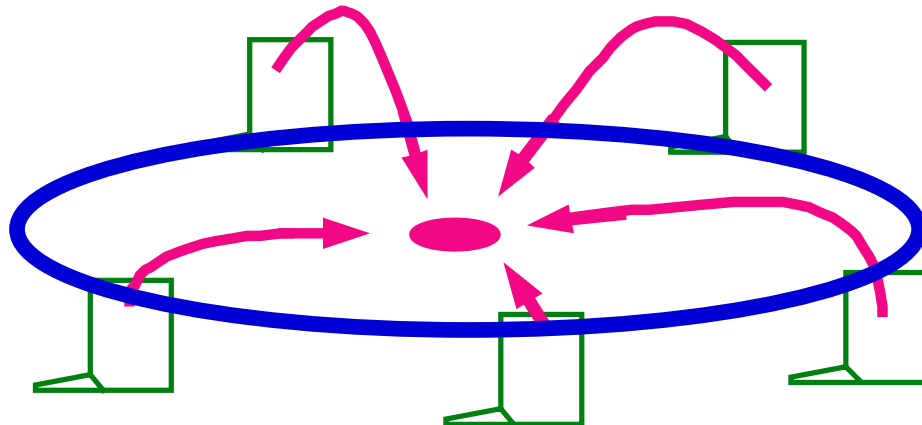
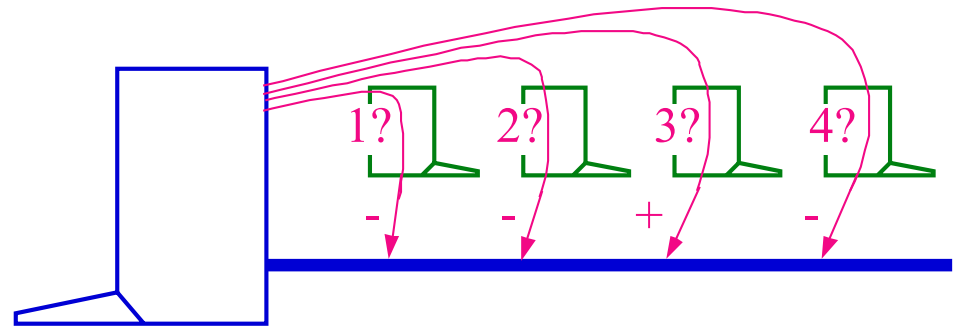
6.8. Adressierung und Gruppierung

- **Namen (mnemonische Funktion):**
 - vs.informatik.uni-ulm.de (Internetname)
 - Peter Schulthess (Personenname)
 - Vertsys_D (Netzbereichsname)
 - pschulth (Benutzername)
 - inputChar (Variablenname)
- **Adressen (Auswahlfunktion):**
 - 00 01 02 A4 B3 C5 (Adapterkarte)
 - 134.60.77.74:80 (Internet adresse)
 - 0049 731/502-4140 (Telefonnummer !)
 - (\$0040:\$001A) (Speicheradresse)
 - \$03f8 (I/O Port)
- **Gruppen-Adressen:**
 - 80 01 02 A4 B3 C5 (Gruppe im Ethernet)
 - FF FF FF FF FF FF (Rundspruch)
 - 0049 731 502 2428 (Modemgruppe am RZ)
- **Namensbindung bedeutet Verbinden eines Namens mit einer Adresse:**
 - statisch Binden(z. Start- oder Übersetzungszeit) oder dynamisch Binden (= zur Laufzeit),
=> Ernst.informatik.uni-ulm.de -> 134.60.77.56

**"Machines use addresses,
people prefer names"**

6.9. Zuteilungsprotokolle

- Wichtig, wenn sich mehrere Stationen eine Leitung teilen.
- Station wird über ihre Adresse angesprochen.
- vgl. Mehrpunktverbindung im Kapitel "Betriebsarten & Verkehrsrichtungen".
- Entweder Abfrage durch zentrale Station:
- Oder dezentrale Zugangsteuerung:
- mit Kollisionsrisiko,
- oder mit Token:



6.10. Verbindungsauf- & Abbau

- Verbindung wird aufgebaut, um Paketstrom als Gesamtheit zu behandeln:
 - Sequenznummern, Flusskontrolle, Dienstgüte ...
- Beim Verbindungsaufbau:
 - Sequenznummern beidseitig initialisieren,
 - Kommunikationspartner identifizieren,
 - Übertragungsweg im Netz suchen,
 - maximale Paketgröße aushandeln,
 - Puffer und Ressourcen allozieren,
 - Dienstgüte aushandeln.
- Beim Verbindungsabbau:
 - Abgrenzung von der nächsten Verbindung,
 - Ressourcen freigeben.
- Eventuell Konferenzverbindungen.
- Eventuell verbindungsloser Dienst:
 - keine Übertragungsgarantien,
 - keine Aufbauverzögerung,
 - keine Sequenznummern.

