

# Systeme II

## 3. Die Datensicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 15.05.2017

- Zumeist gefordert von der Vermittlungsschicht
  - Mit Hilfe der Frames
- Fehlererkennung
  - Gibt es fehlerhaft übertragene Bits?
- Fehlerkorrektur
  - Behebung von Bitfehlern
  - Vorwärtsfehlerkorrektur (Forward Error Correction)
    - Verwendung von redundanter Kodierung, die es ermöglicht Fehler ohne zusätzliche Übertragungen zu beheben
  - Rückwärtsfehlerkorrektur (Backward Error Correction)
    - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben

## Fehlerkontrolle

Fehlererkennung

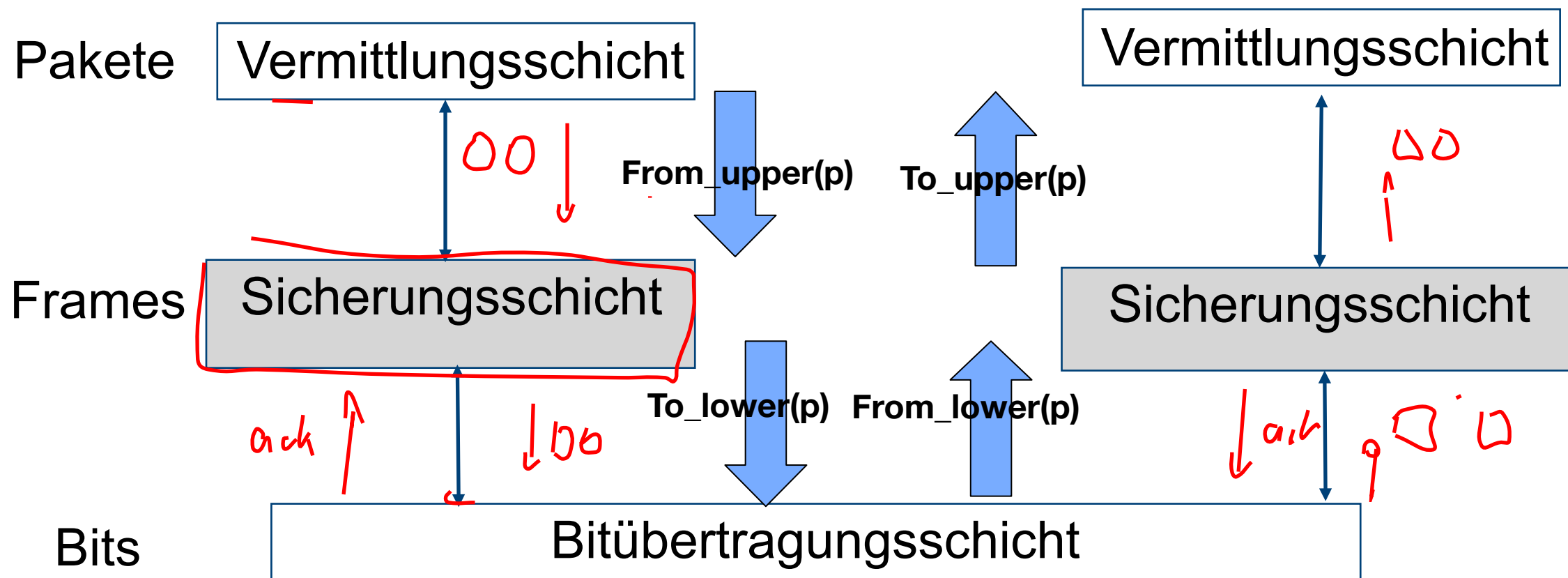
Fehlerkorrektur

Vorwärtsfehler-  
korrektur

Rückwärtsfehler-  
korrektur

# Rückwärtsfehlerkorrektur

- Bei Fehlererkennung muss der Frame nochmal geschickt werden
- Wie ist das Zusammenspiel zwischen Sender und Empfänger?

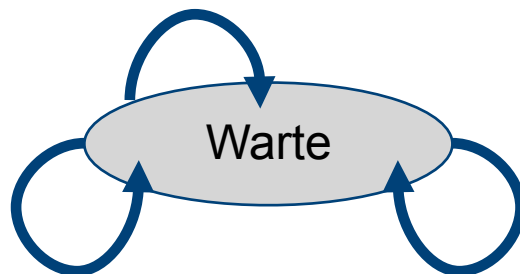


to\_lower, from\_lower beinhalten CRC  
oder (bei Bedarf) Vorwärtsfehlerkorrektur

- Empfänger bestätigt Pakete dem Sender
- Der Sender wartet für eine bestimmte Zeit auf die Bestätigung (acknowledgment)
- Falls die Zeit abgelaufen ist, wird das Paket wieder versendet
- Erster Lösungsansatz

## Sender

From\_upper (p);  
set\_timer, to\_lower(p)

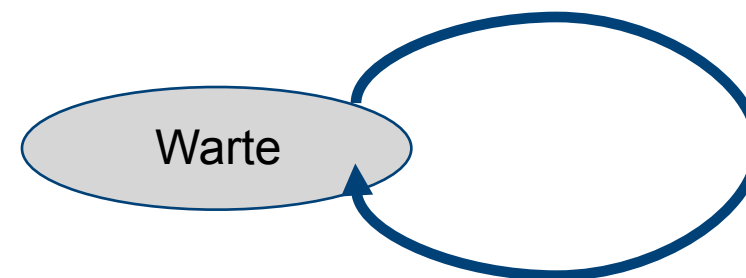


From\_lower (ack);  
cancel\_timer

timeout;  
to\_lower (p),  
set\_timer

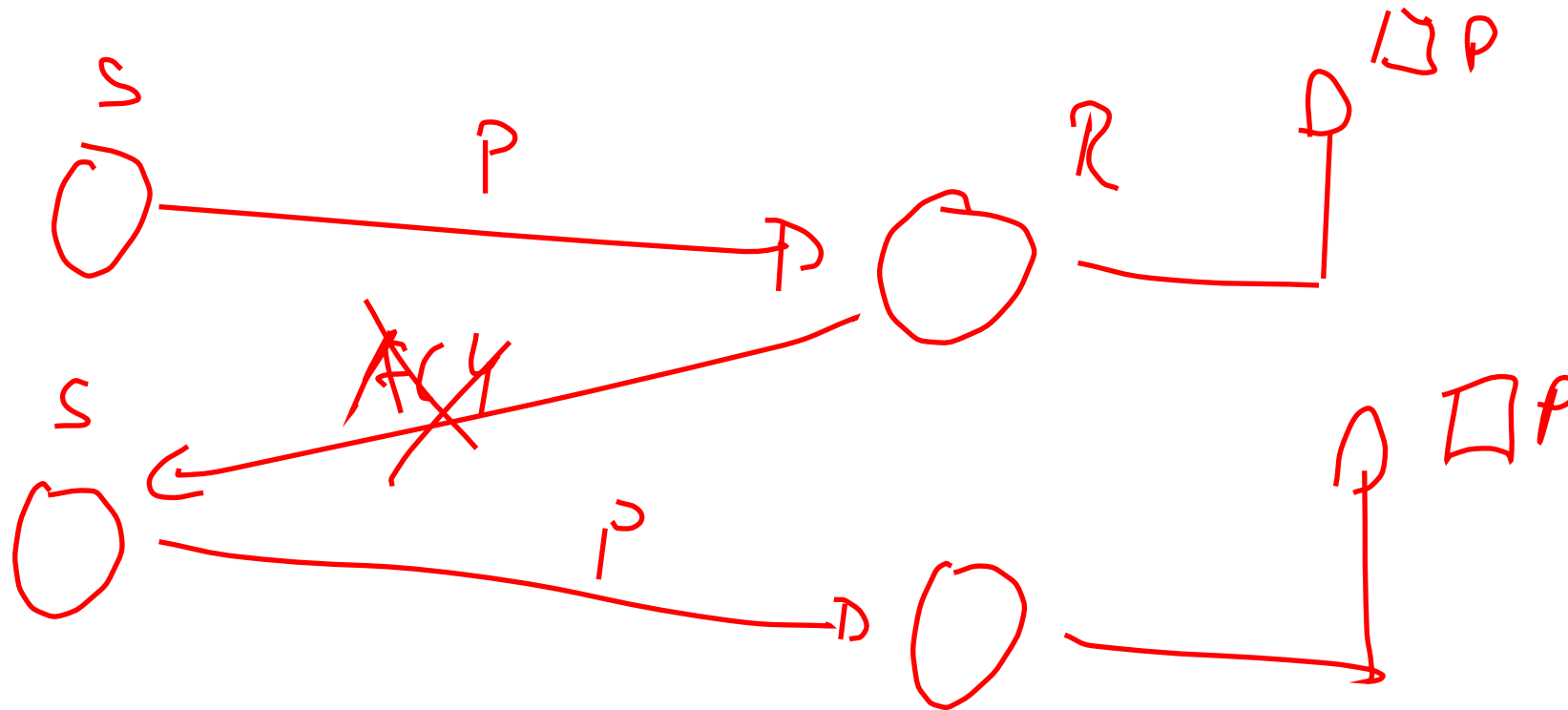
## Empfänger

From\_lower (p);  
To\_upper(p),  
To\_lower (ack)



## ■ Probleme

- Sender ist schneller als Empfänger
- Was passiert, wenn Bestätigungen verloren gehen?

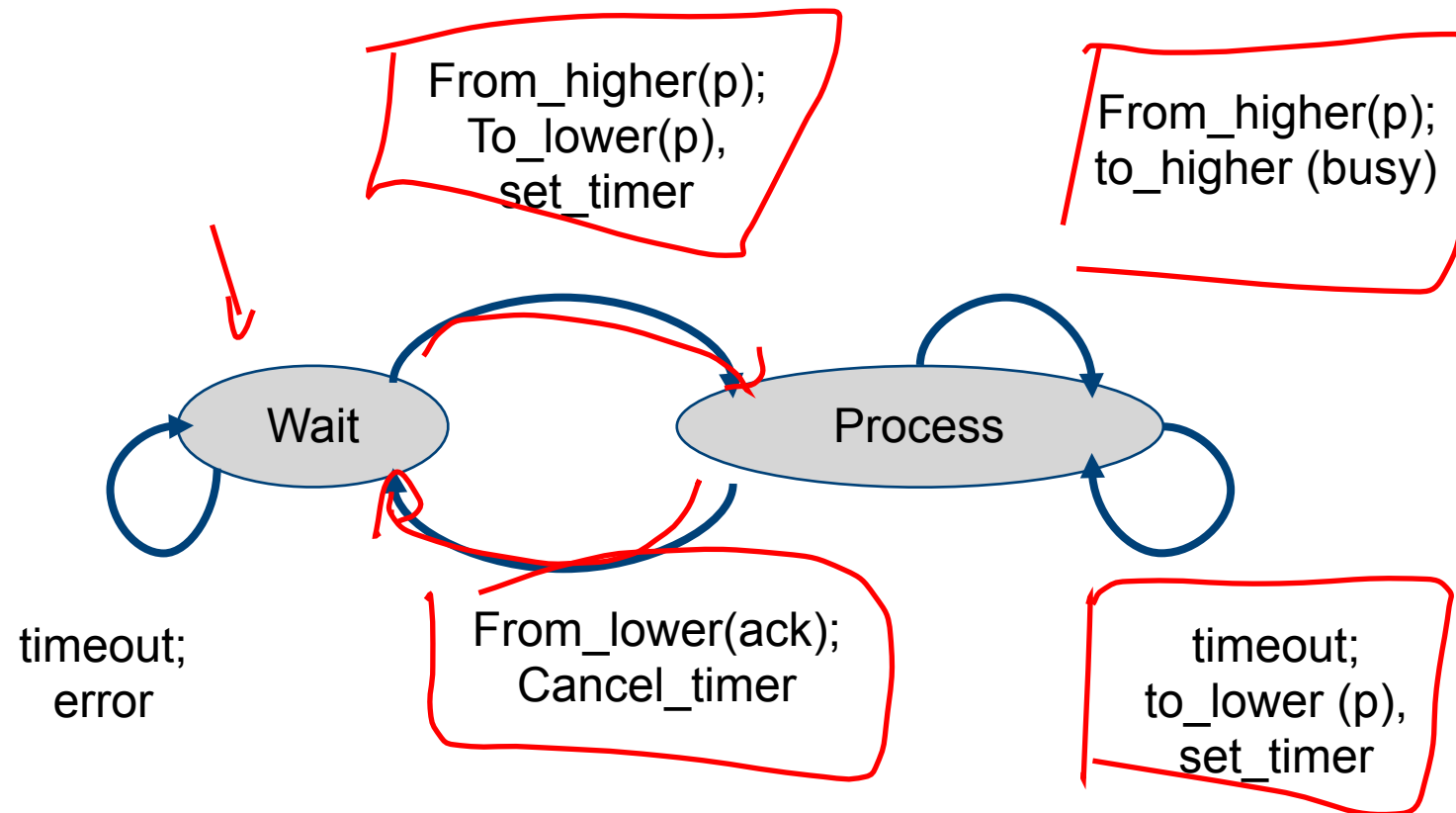


## 2. Versuch

### ■ Lösung des ersten Problems

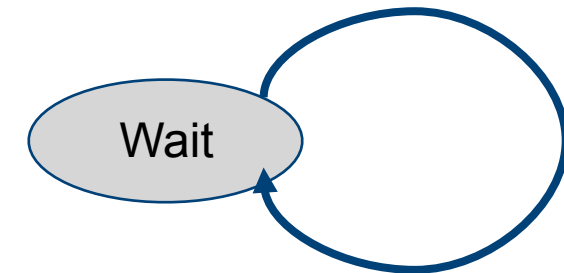
- Ein Paket nach dem anderen

#### - Sender

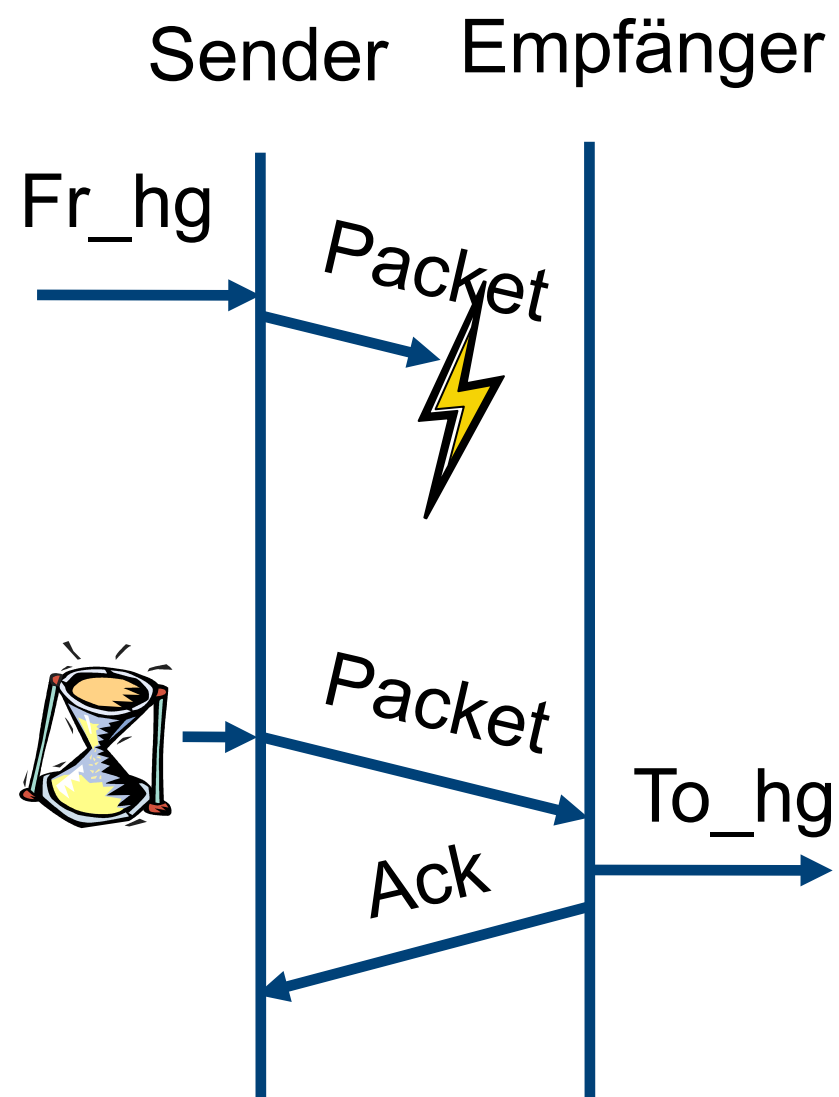
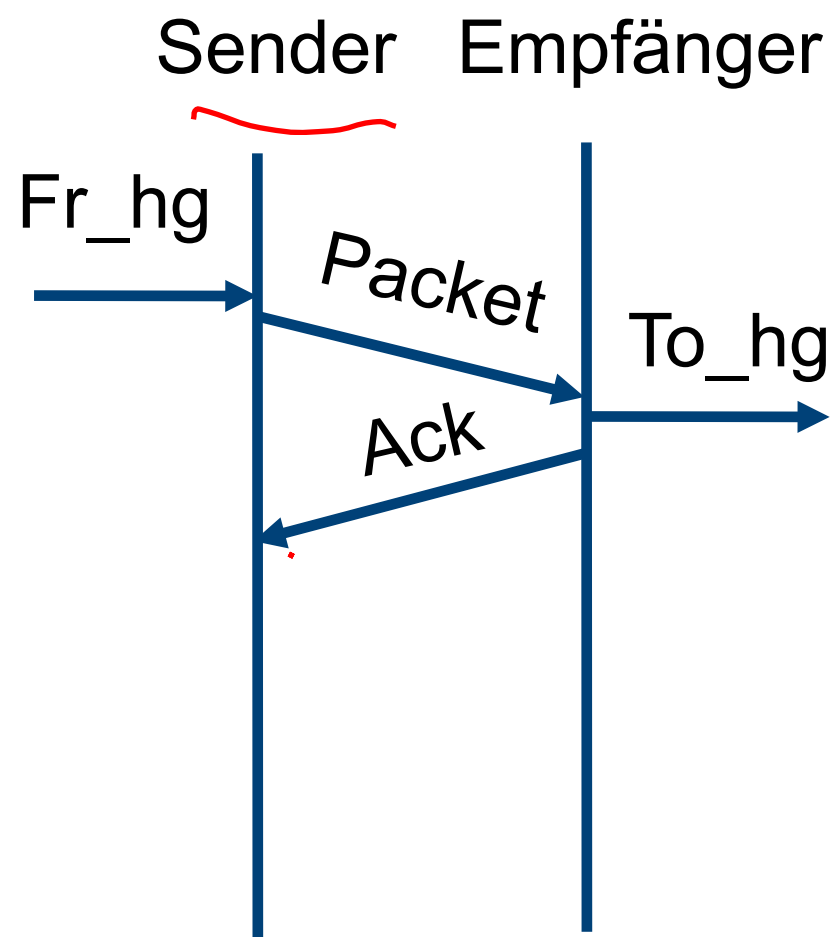


#### Empfänger

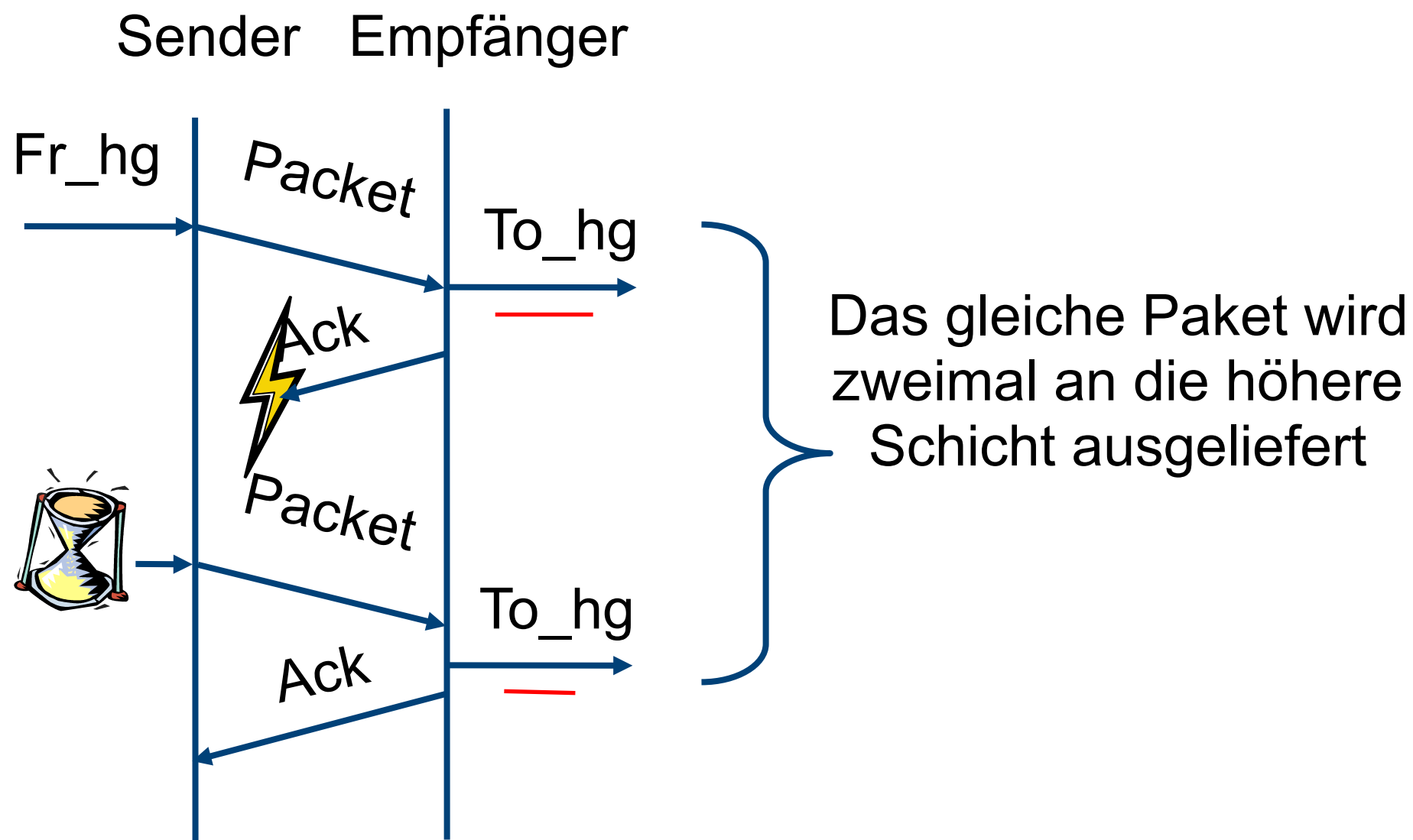
From\_lower (p);  
To\_upper(p),  
to\_lower (ack)



- Protokoll etabliert elementare Flusskontrolle



- 2. Fall: Verlust von Bestätigung



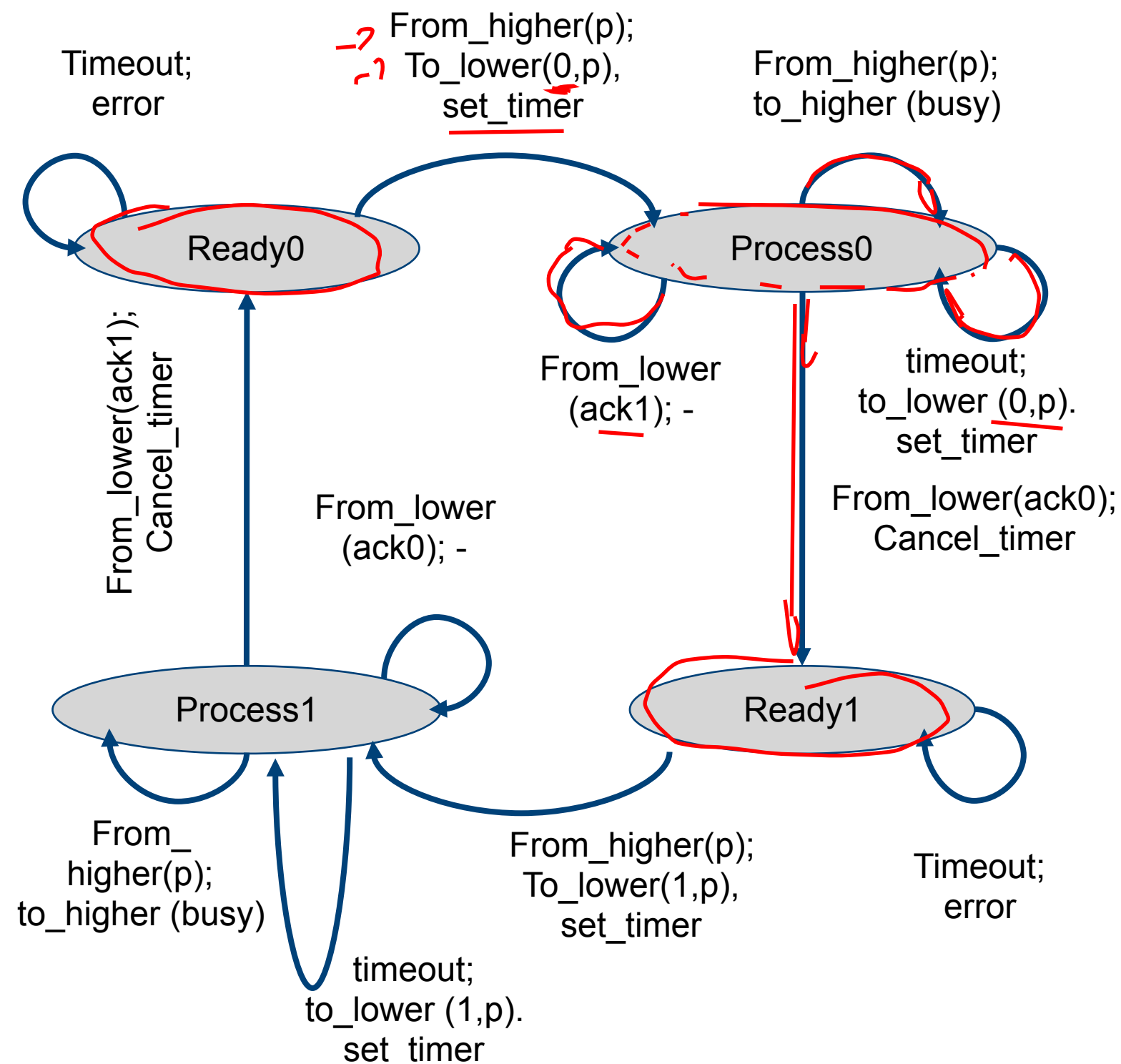


# Probleme der 2. Version

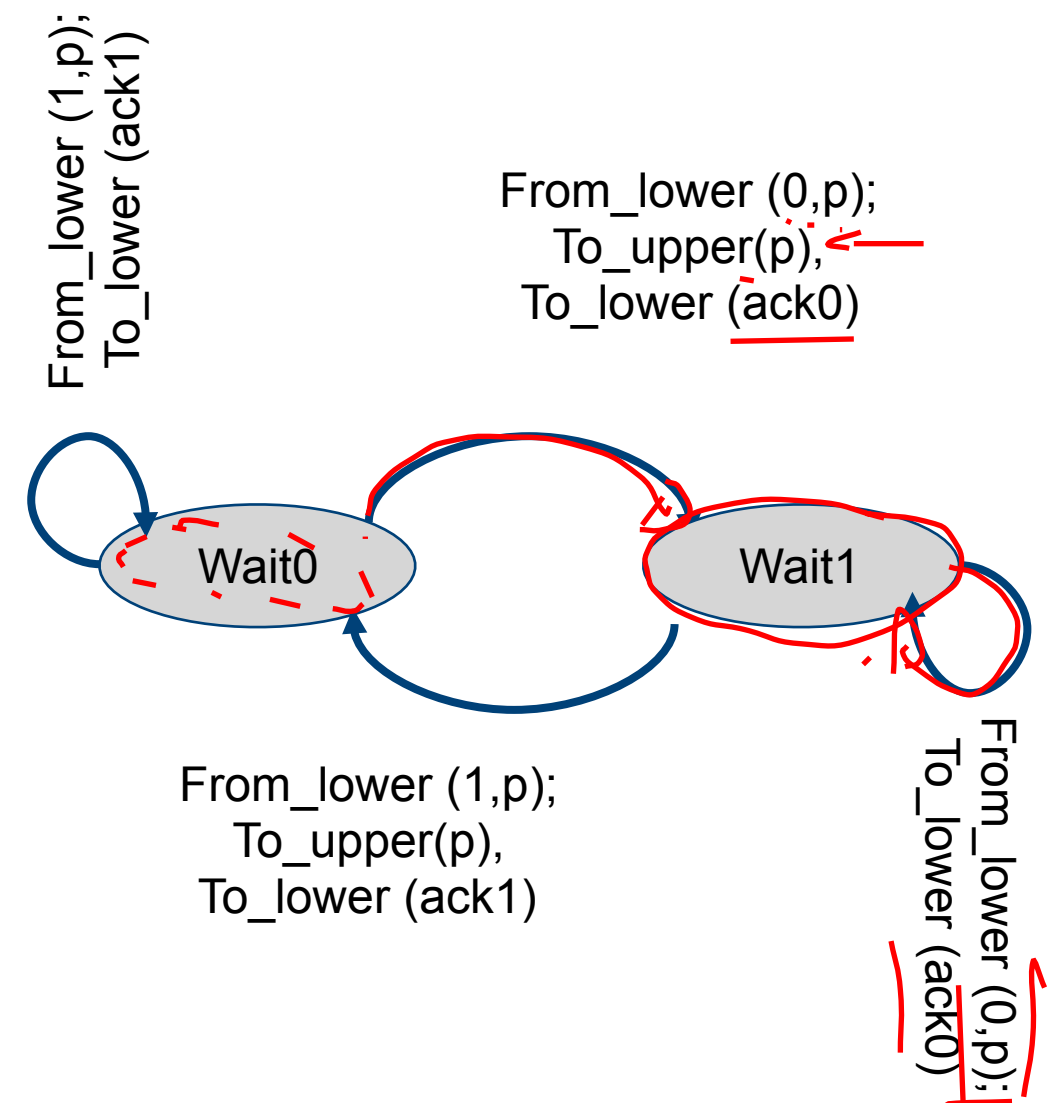
- Sender kann nicht zwischen verlorenem Paket und verllorener Bestätigung unterscheiden
  - Paket muss neu versendet werden
- Empfänger kann nicht zwischen Paket und redundanter Kopie eines alten Pakets unterscheiden
  - Zusätzliche Information ist notwendig
- Idee:
  - Einführung einer Sequenznummer in jedes Paket, um den Empfänger Identifikation zu ermöglichen
  - Sequenznummer ist im Header jedes Pakets
  - Hier: nur 0 oder 1
- Notwendig in Paket und Bestätigung
  - In der Bestätigung wird die Sequenznummer des letzten korrekt empfangenen Pakets mitgeteilt
    - (reine Konvention)

# 3. Versuch: Bestätigung und Sequenznummern

## Sender



## Empfänger



# 3. Version

## Alternating Bit Protocol

---

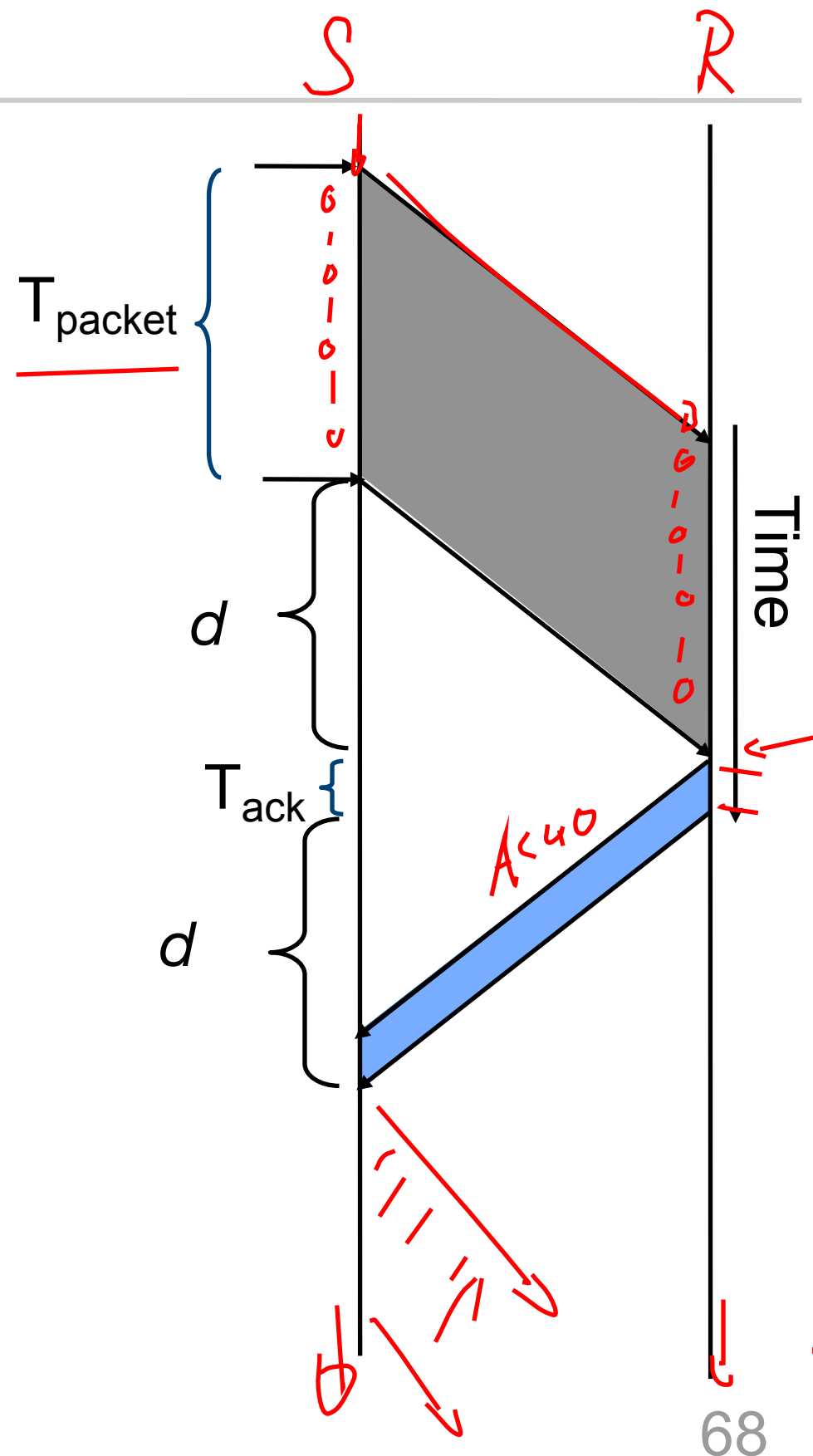
- Die 3. Version ist eine korrekte Implementation eines verlässlichen Protokolls über einen gestörten Kanal
  - Alternating Bit Protokoll
  - aus der Klasse der Automatic Repeat reQuest (ARQ) Protokolle
  - beinhaltet auch eine einfache Form der Flusskontrolle
- Zwei Aufgaben einer Bestätigung
  - ⌚ Bestätigung, dass Paket angekommen ist
  - ⌚ Erlaubnis ein neues Paket zu schicken

## ■ Effizienz $\eta$

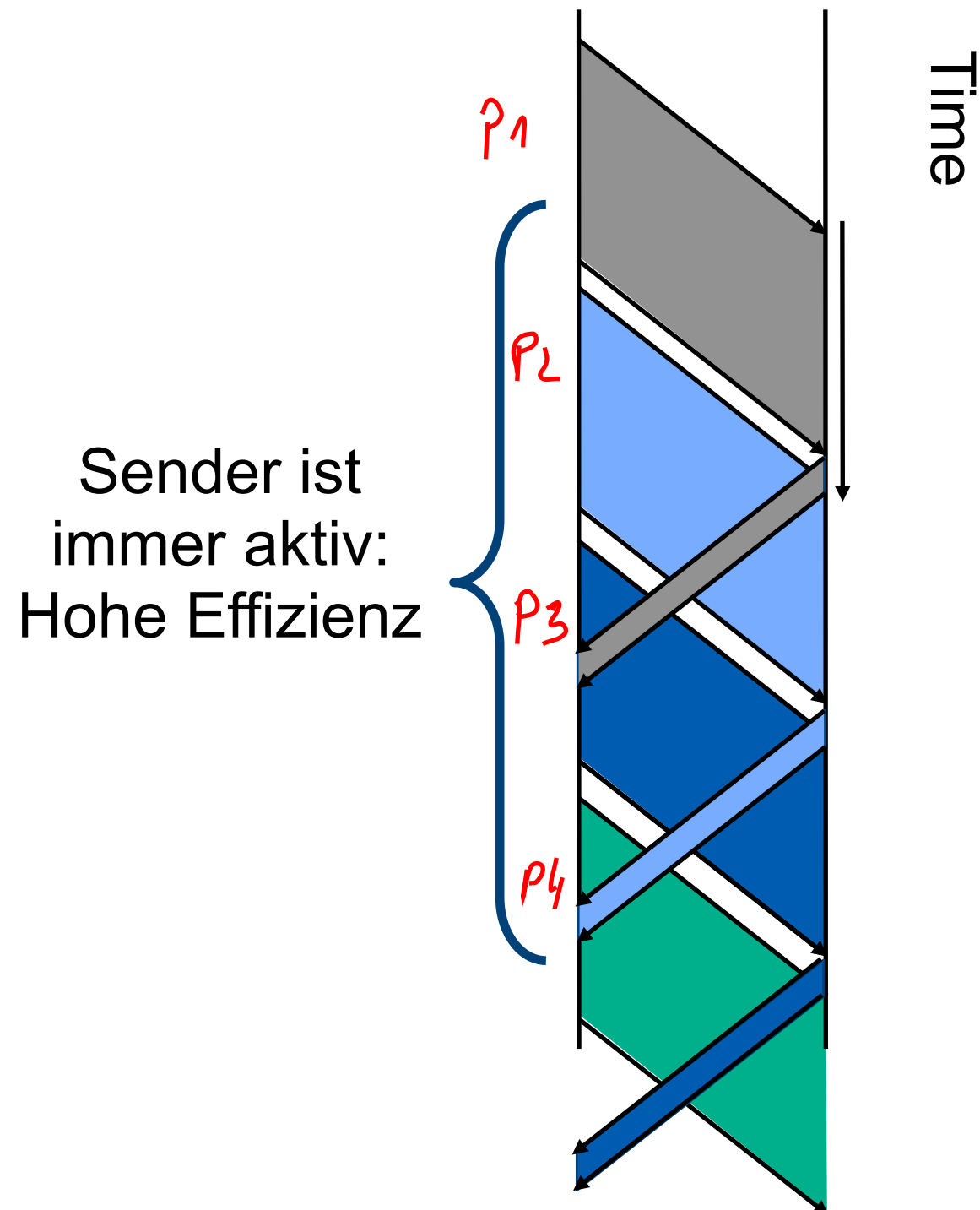
- Definiert als das Verhältnis zwischen
  - der Zeit um zu senden
  - und der Zeit bis neue Information gesendet werden kann
  - (auf fehlerfreien Kanal)

$$\eta = \frac{T_{\text{packet}}}{T_{\text{packet}} + d + T_{\text{ack}} + d}$$

- Bei großen Delay ist das Alternating Bit Protocol nicht effizient



- Durchgehendes Senden von Paketen erhöht Effizienz
  - Mehr “ausstehende” nicht bestätigte Pakete erhöhen die Effizienz
  - “Pipeline” von Paketen
- Nicht mit nur 1-Bit-Sequenznummer möglich



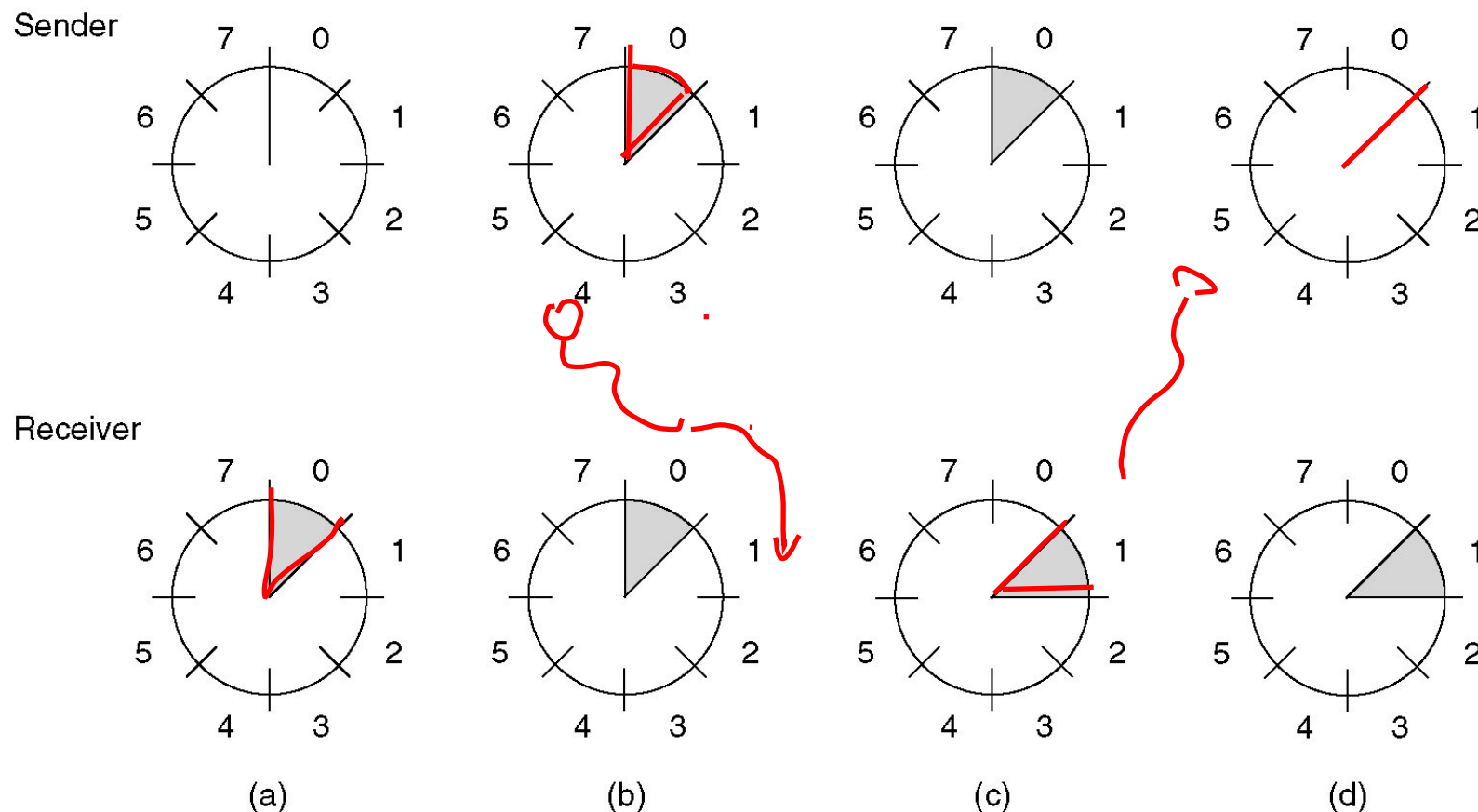
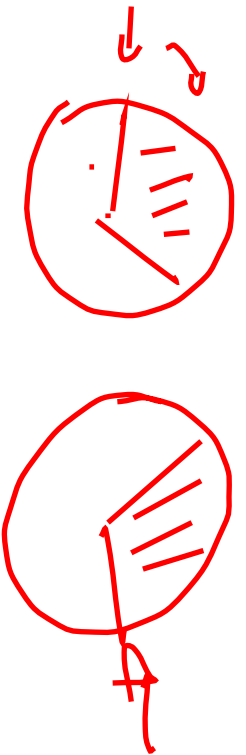
# Gleitende Fenster

0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7  
□ □ □

- Der Raum für Sequenznummern wird vergrößert
  - auf n Bits oder  $2^n$  Sequenznummern
- Nicht alle davon können gleichzeitig verwendet werden
  - auch bei Alternating Bit Protocol nicht möglich
- 0 ■ “Gleitende Fenster” (sliding windows) bei Sender und Empfänger behandeln dieses Problem
  - Sender: Sende-Fenster
    - Folge von Sequenznummer, die zu einer bestimmten Zeit gesendet werden können
  - Empfänger: Empfangsfenster
    - Folge von Sequenznummer, die er zu einer bestimmten Zeit zu akzeptieren bereit ist
  - Größe der Fenster können fest sein oder mit der Zeit verändert werden
  - Fenstergröße entspricht Flusskontrolle

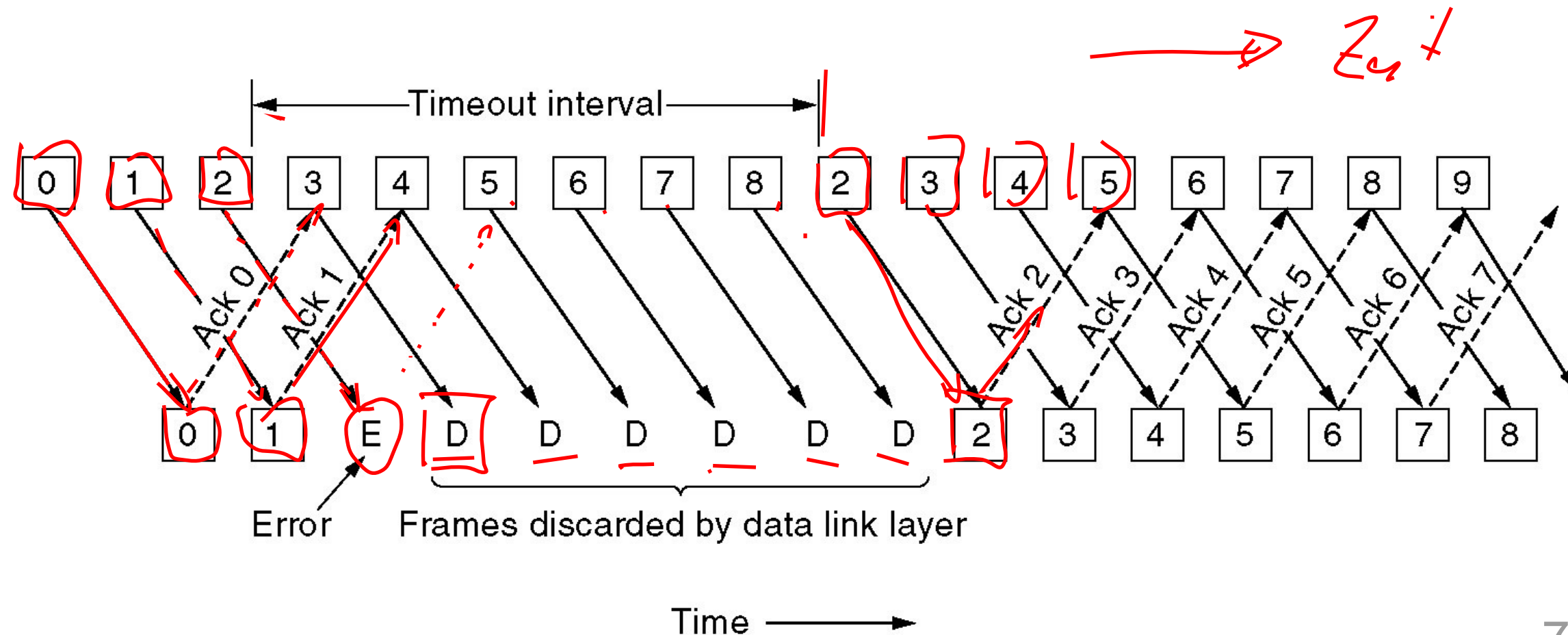
$$2^3 = 8$$

- “Sliding Window”-Beispiel für  $n=3$  und fester Fenstergröße = 1
- Der Sender zeigt die momentan unbestätigten Sequenznummern an
  - Falls die maximale Anzahl nicht bestätigter Frames bekannt ist, dann ist das das Sende-Fenster



- Initial: Nichts versendet
- Nach Senden des 1. Frames mit Seq.Nr. 0
- Nach dem Empfang des 1. Frame
- Nach dem Empfang der Bestätigung

- Annahme:
  - Sicherungsschicht muss alle Frames korrekt in der richtigen Reihenfolge verschicken
  - Sender “pipelined” Paket zur Erhöhung der Effizienz
- Bei Paketverlust:
  - werden alle folgenden Pakete ebenfalls fallen gelassen





- Mit Empfangsfenster der Größe 1 können die Frames, die einem verlorenen Frame folgen, nicht durch den Empfänger bearbeitet werden
  - Sie können einfach nicht bestätigt werden, da nur eine Bestätigung für das letzte korrekt empfangene Paket verschickt wird
- Der Sender wird einen “Time-Out” erhalten
  - Alle in der Zwischenzeit versandten Frames müssen wieder geschickt werden
  - “Go-back N” Frames!
- Kritik
  - Unnötige Verschwendung des Mediums
  - Spart aber Overhead beim Empfänger

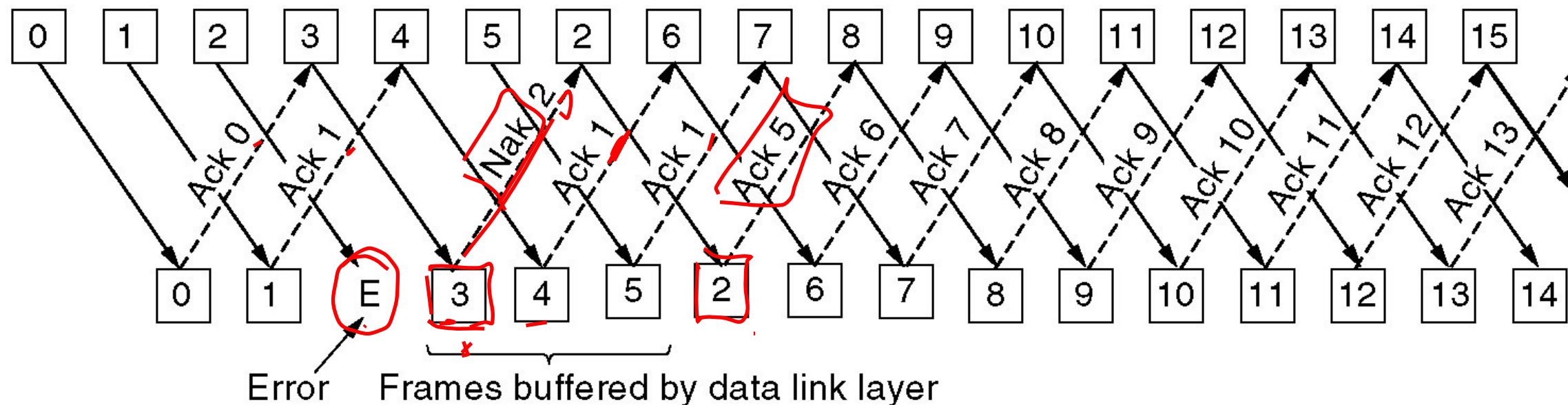
# Selektierte Wiederholung

0 1 2 3 4 5

## ■ Angenommen

- der Empfänger kann die Pakete puffern, welche in der Zwischenzeit angekommen sind
- d.h. das Empfangsfenster ist größer als 1

## ■ Beispiel

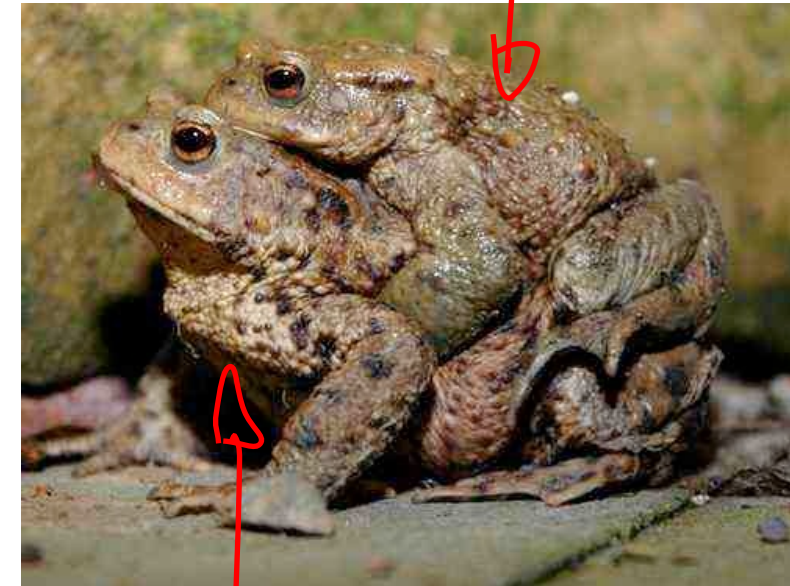


- Der Empfänger informiert dem Sender fehlende Pakete mit negativer Bestätigung
- Der Sender verschickt die fehlenden Frames selektiv
- Sobald der fehlende Frame ankommt, werden alle (in der korrekten Reihenfolge) der Vermittlungsschicht übergeben

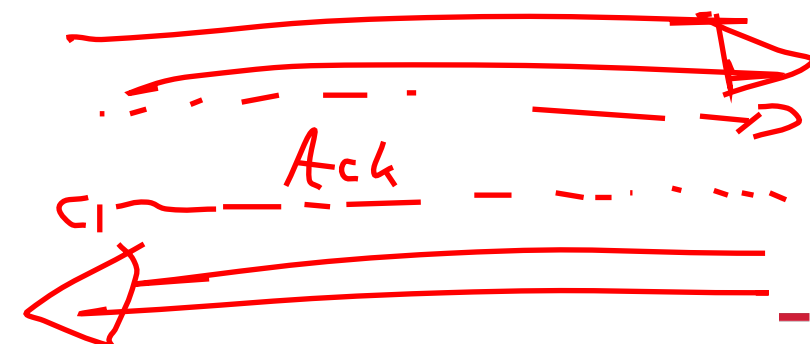
$A \rightarrow B$

$Ack^{ve} A \rightarrow B$

- Simplex
  - Senden von Informationen in einer Richtung
- Duplex
  - Senden von Informationen in beide Richtungen
- Bis jetzt:
  - Simplex in der Vermittlungsschicht
  - Duplex in der Sicherungsschicht
- Duplex in den höheren Schichten
  - Nachrichten und Datenpakete separat in jeder Richtung
  - Oder Rucksack-Technik
    - Die Bestätigung wird im Header eines entgegen kommenden Frames gepackt



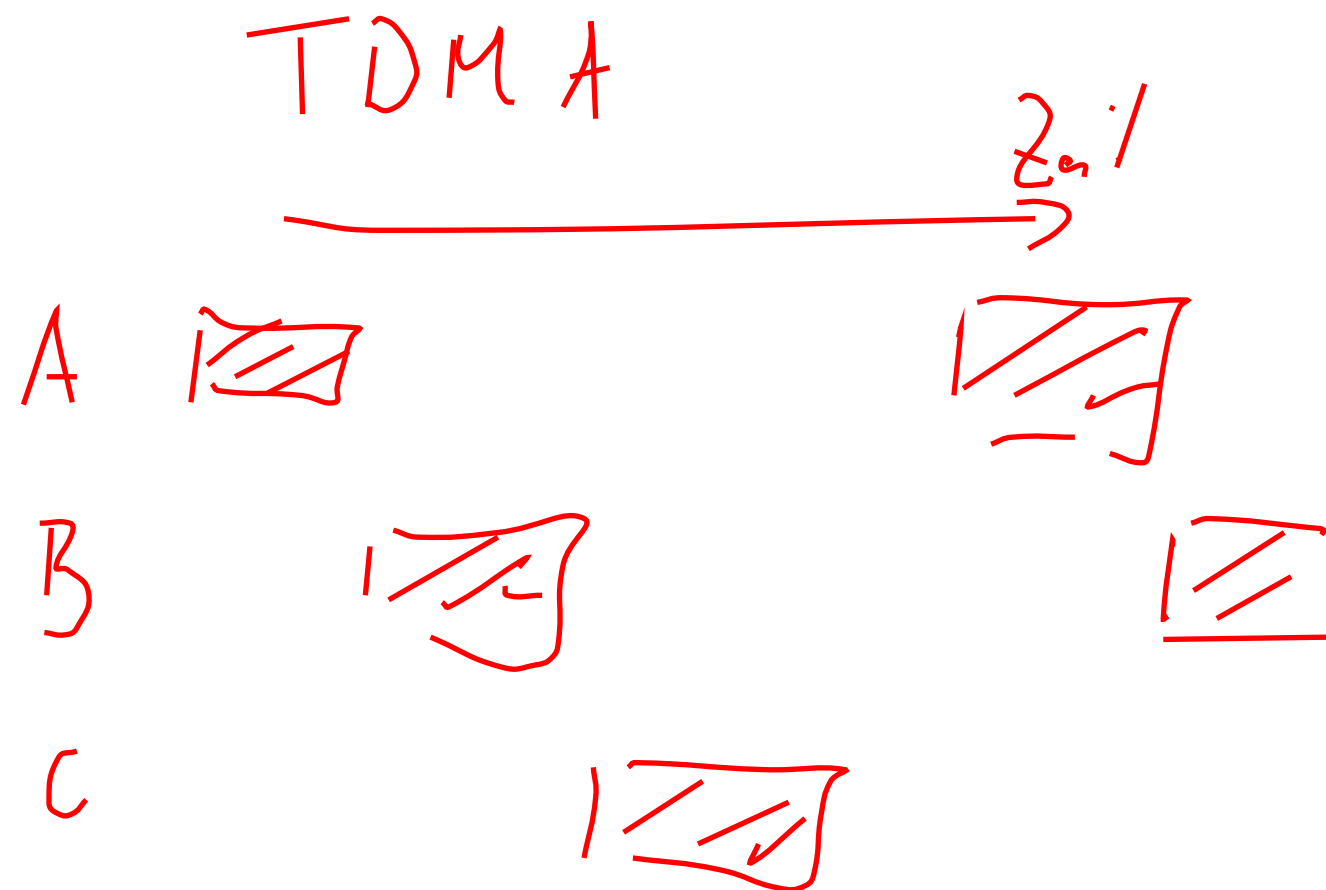
$A \quad B$   
Nachricht  $A \leftarrow B$



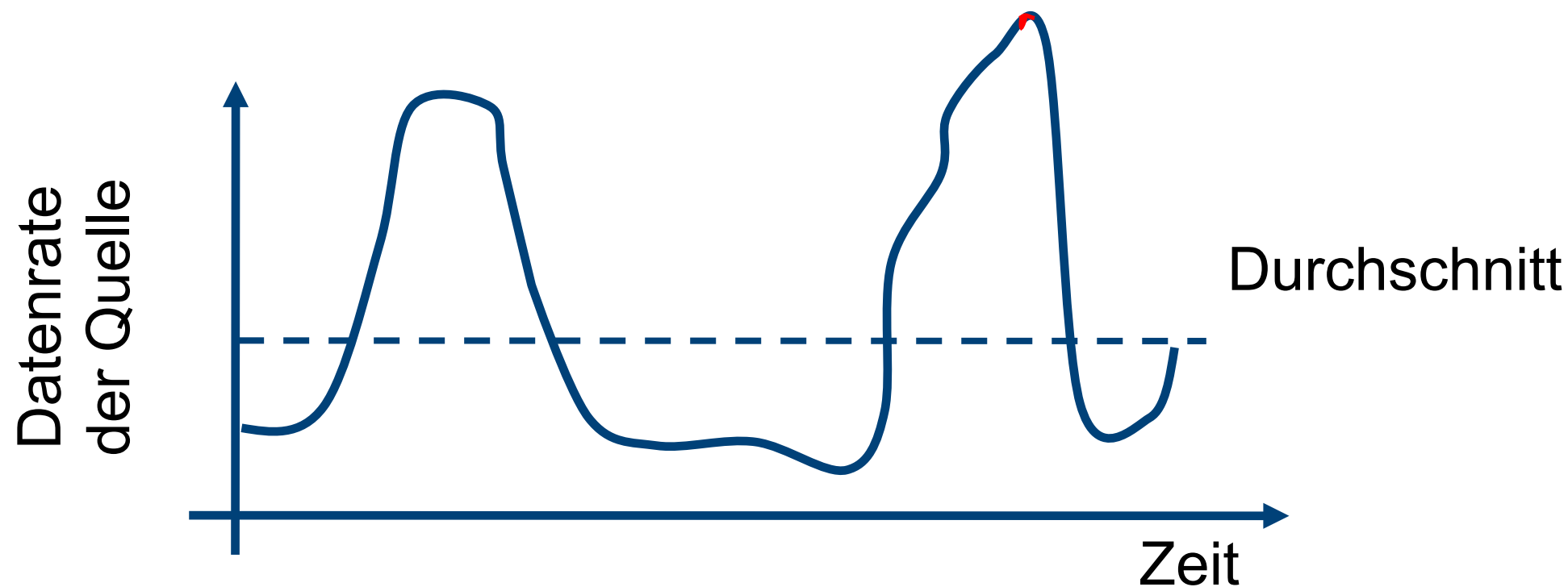
- Die Bitübertragung kann erst stattfinden, wenn das Medium reserviert wurde
  - Funkfrequenz bei drahtloser Verbindung (z.B. W-LAN 802.11, GSM, GPRS)
  - Zeitraum bei einem Kabel mit mehreren Rechnern (z.B. Ethernet)
- Aufgabe der Sicherungsschicht
  - Koordination zu komplex für die “einfache” Bitübertragungsschicht

- Statisches Multiplexen
- Dynamische Kanalbelegung
  - Kollisionsbasierte Protokolle
  - Kollisionsfreie Protokolle (contention-free)
  - Protokolle mit beschränktem Wettbewerb (limited contention)

- Gegeben sei eine einzelne Leitung (Ressource)
- Mehreren Kommunikationsverbindungen werden feste Zeiträume/Kanäle (slots/channels) zugewiesen
  - Oder: Feste Frequenzbänder werden ihnen zugewiesen
- Feste Datenraten und entsprechenden Anteilen am Kanal
  - Quellen lasten die Leitung aus



- Problem: Verkehrsspitzen (bursty traffic)
  - Definition: Großer Unterschied zwischen Spitze und Durchschnitt
  - In Rechnernetzwerken: Spitze/Durchschnitt = 1000/1 nicht ungewöhnlich



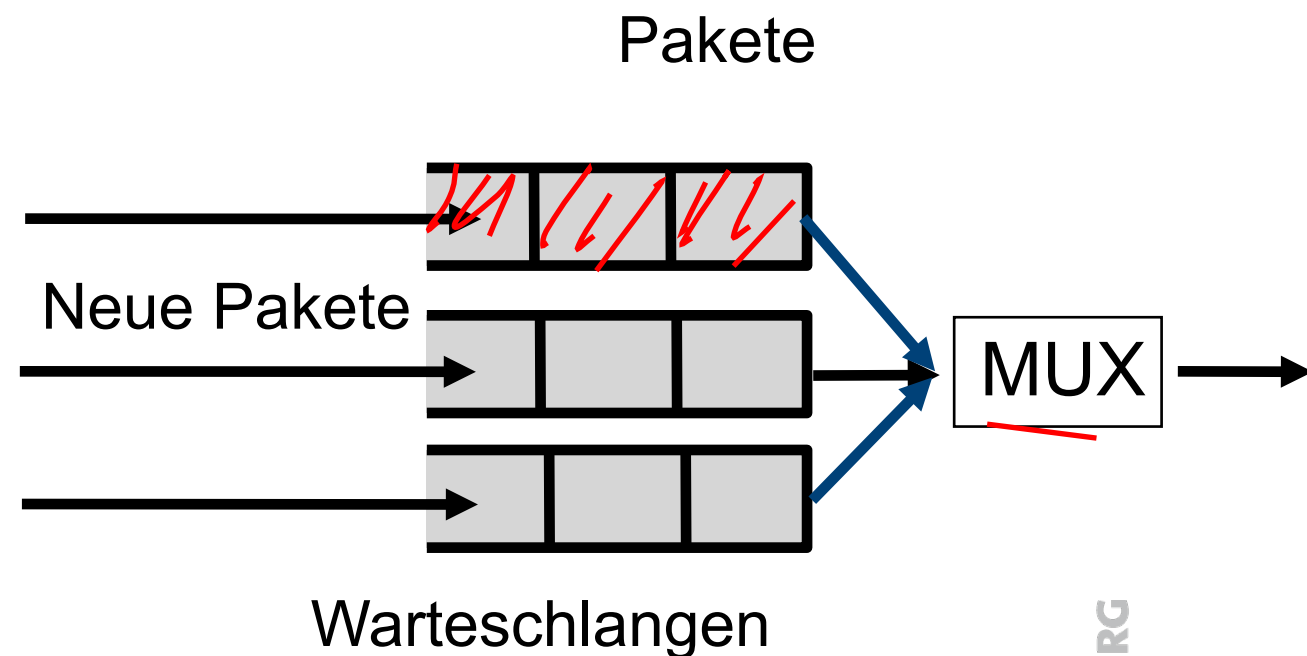
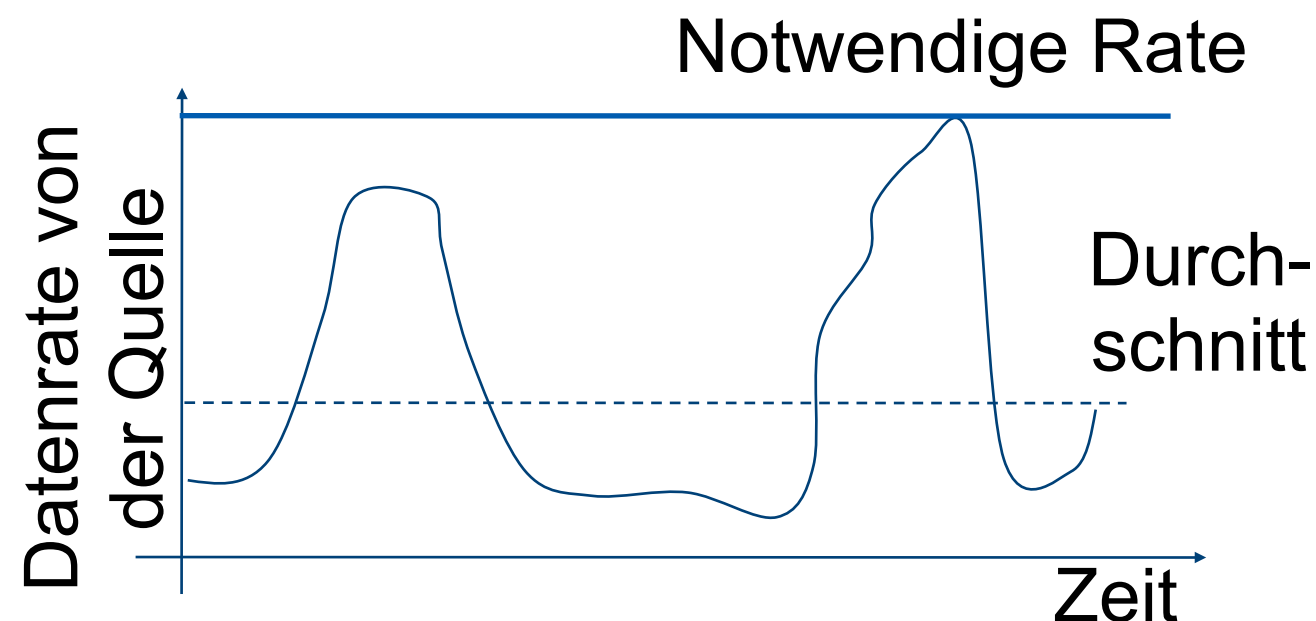
- Leitung für statisches Multiplexen:

- entweder

- Genügend große Kapazität um mit dem Peak fertig zu werden
- Verschwendung, da die Durchschnittsrate den Kanal nicht auslasten wird

- oder

- Ausgelegt für Durchschnittsrate
- Versehen mit Warteschlangen (queue)
- Vergrößerung der Verzögerung (delay) der Pakete





- Vergleich der Verzögerung
- Ausgangsfall:
  - Kein Multiplexing
  - Einfacher Datenquelle mit Durchschnittsrate  $\rho$  (bits/s) und der Leitungskapazität  $C$  bits/s
  - Sei  $T$  die Verzögerung
- Multiplex-Fall
  - Die Datenquelle wird in  $N$  Quellen unterteilt mit der selben Datenrate
  - Statischer Multiplex über die selbe Leitung
  - Dann ergibt sich (im wesentlichen) die Verzögerung:  $N T$
- Schluss: Statisches Multiplexen vergrößert den Delay eines Pakets in der Regel um den Faktor  $N$ 
  - Grund: Bei einer Verkehrsspitze sind  $n-1$  Kanäle leer