

Systeme II



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Sommersemester 2007
6. Vorlesungswoche
21.05.-25.05.2007
schindel@informatik.uni-freiburg.de

Systeme II
Kapitel 3
Sicherungsschicht

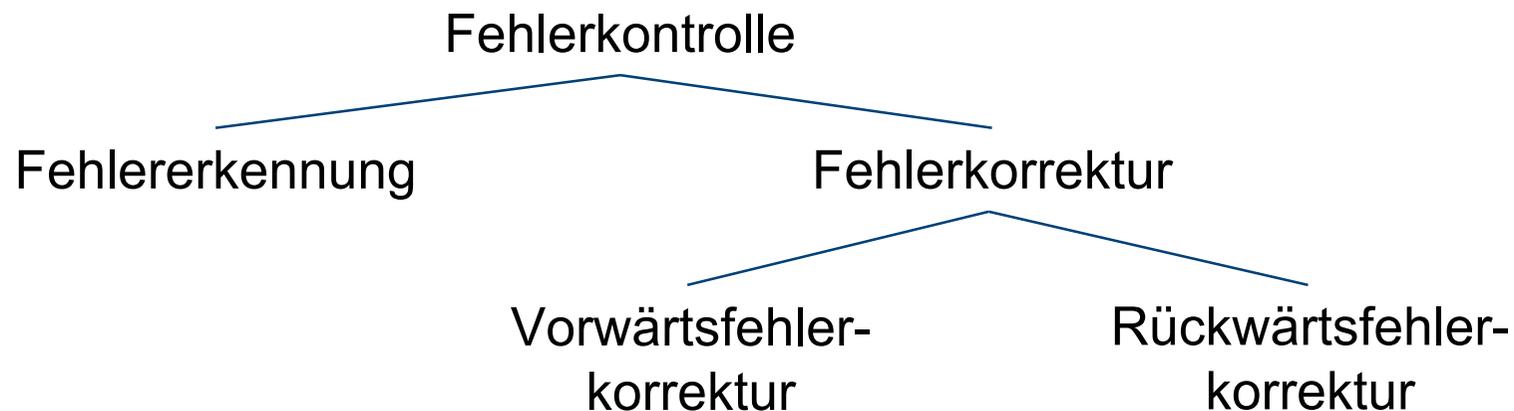


Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Fehlerkontrolle

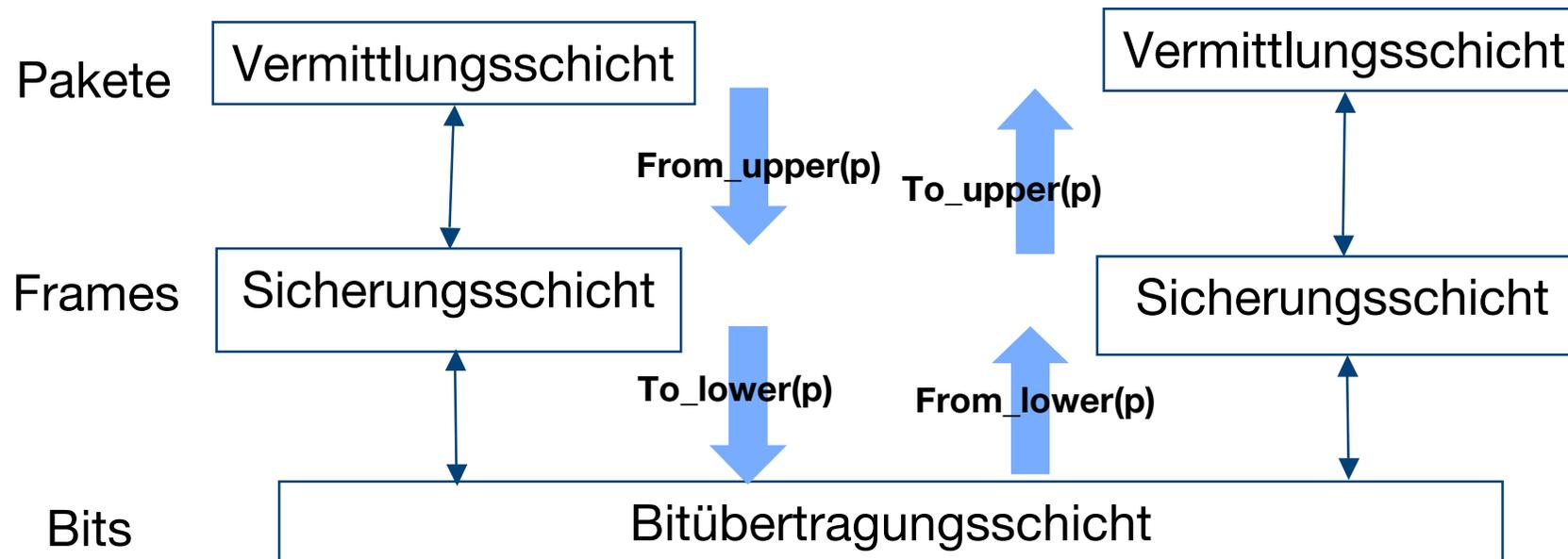
- **Zumeist gefordert von der Vermittlungsschicht**
 - Mit Hilfe der Frames
- **Fehlererkennung**
 - Gibt es fehlerhaft übertragene Bits?
- **Fehlerkorrektur**
 - Behebung von Bitfehlern
 - Vorwärtsfehlerkorrektur (Forward Error Correction)
 - Verwendung von redundanter Kodierung, die es ermöglicht Fehler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





Rückwärtsfehlerkorrektur

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

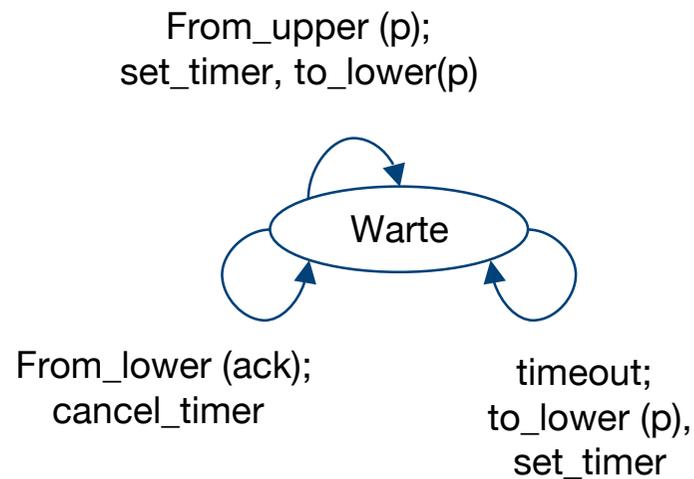


to_lower, from_lower beinhalten CRC
oder (bei Bedarf) Vorwärtsfehlerkorrektur

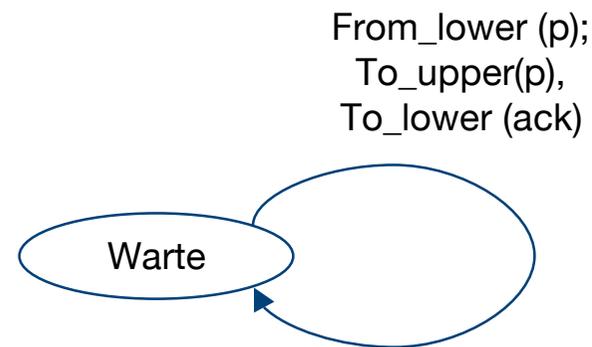


Einfaches Simplex-Protokoll mit Bestätigungen

- **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**



Empfänger





Diskussion

➤ Probleme

– Sender ist schneller als Empfänger

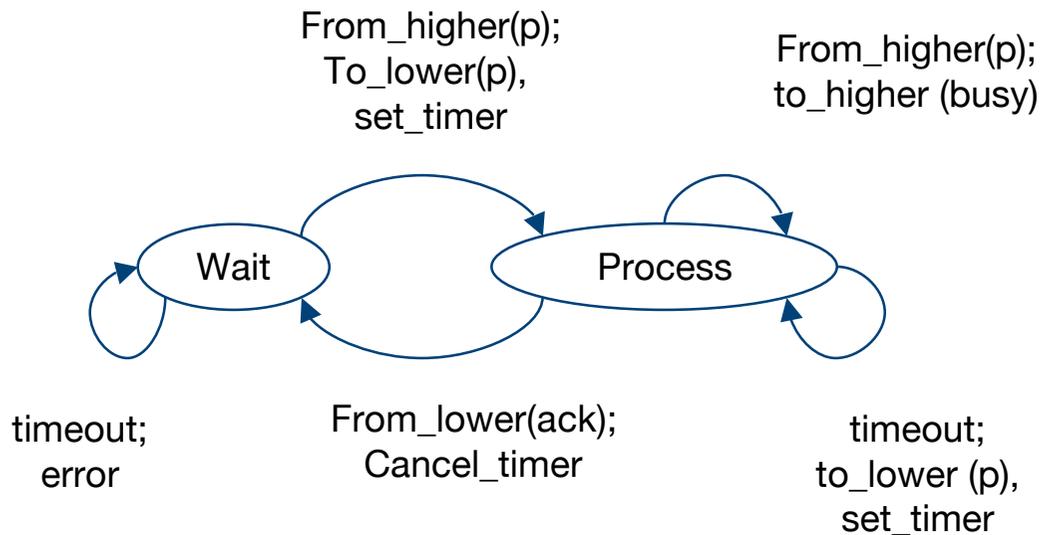
– Was passiert, wenn Bestätigungen verloren gehen?



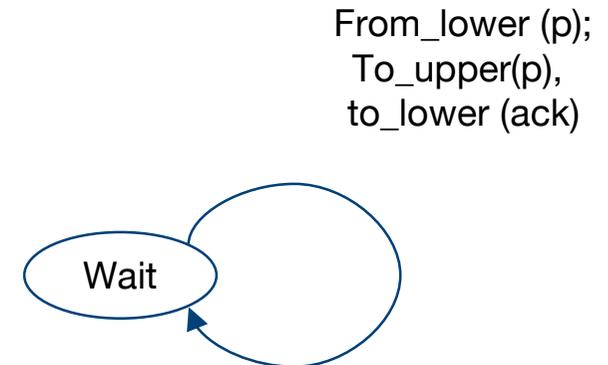
2. Versuch

- Lösung des ersten Problems
 - Ein Paket nach dem anderen

Sender



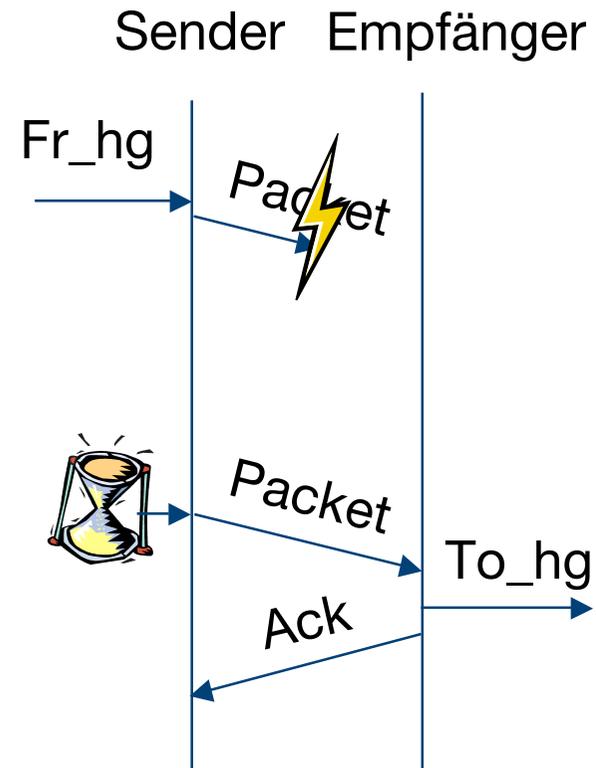
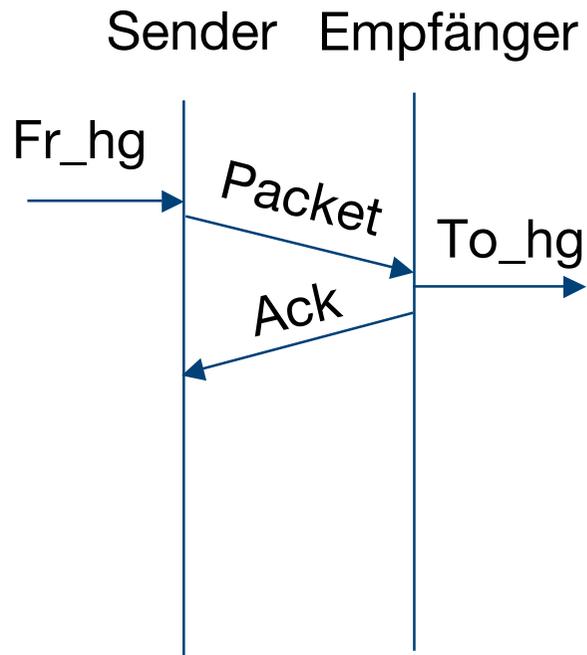
Empfänger





Diskussion

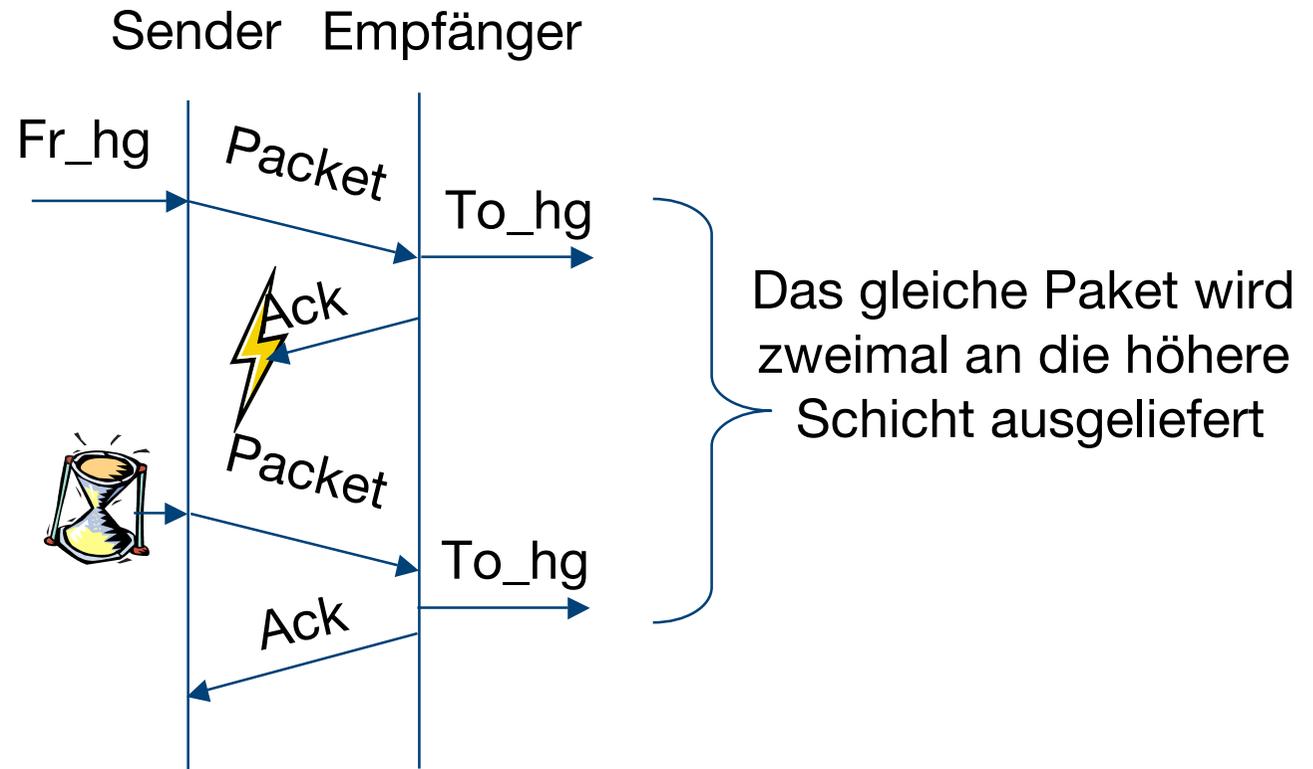
➤ Protokoll etabliert elementare Flusskontrolle





Diskussion

➤ 2. Fall: Verlust von Bestätigung



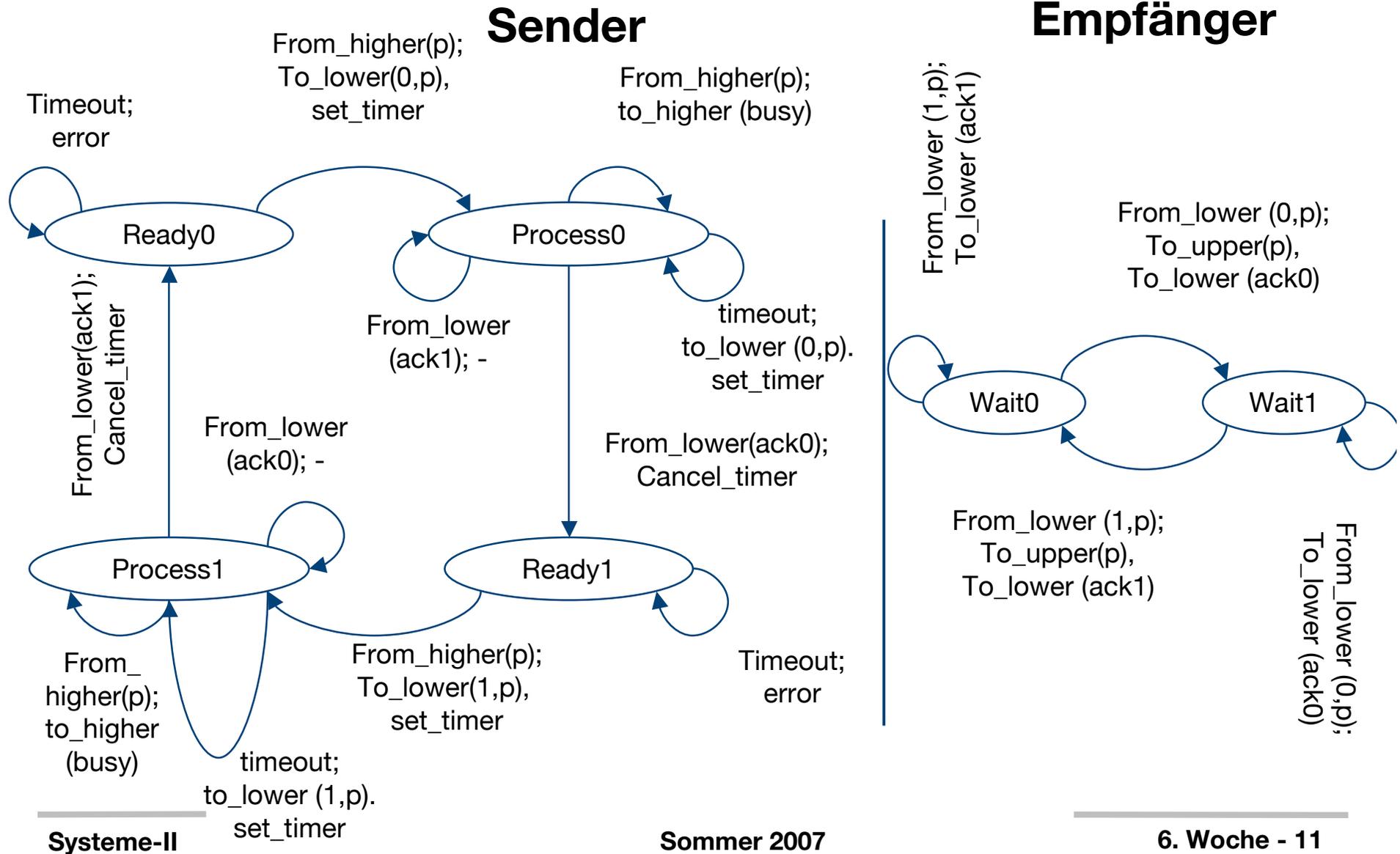


Probleme der 2. Version

- **Sender kann nicht zwischen verlorenem Paket und verlorener 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





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



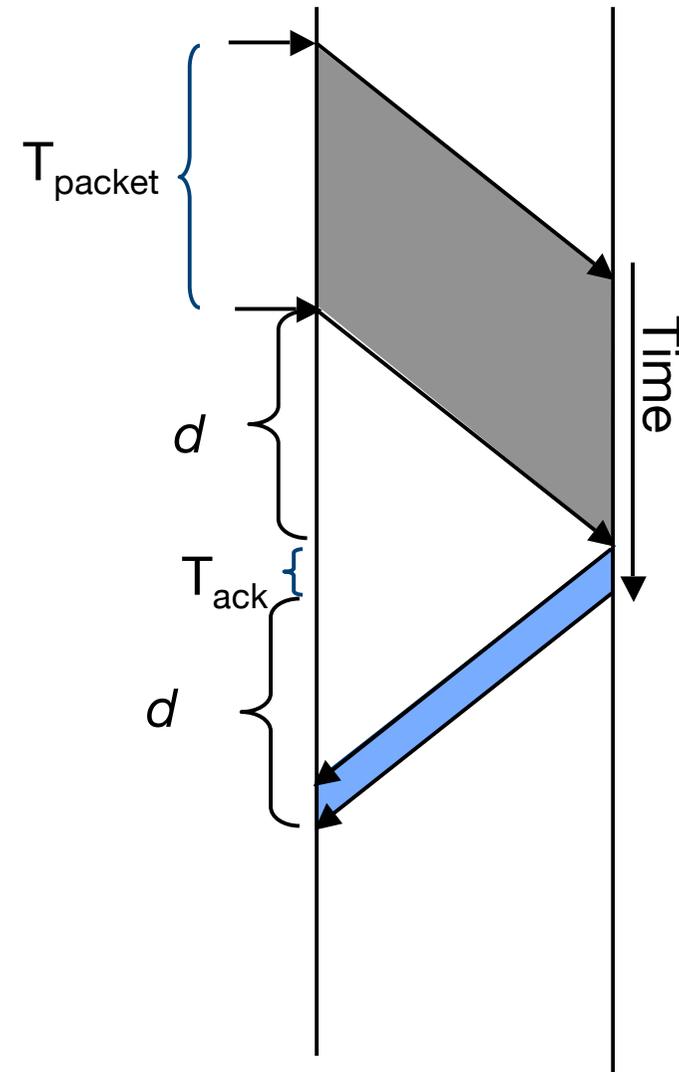
Alternating Bit Protocol – Effizienz

➤ Effizienz η

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

$$\eta = T_{\text{packet}} / (T_{\text{packet}} + d + T_{\text{ack}} + d)$$

➤ Bei großen Delay ist das Alternating Bit Protocol nicht effizient

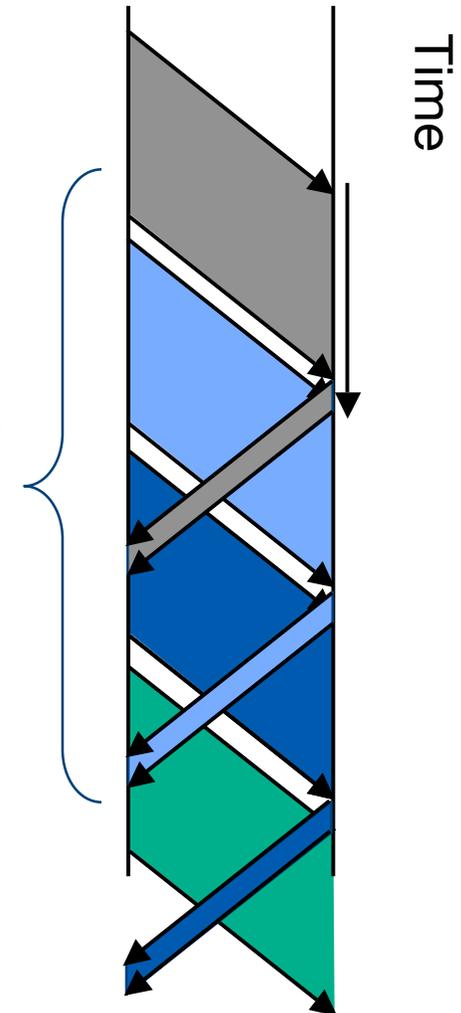




Verbesserung der Effizienz

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

Sender ist immer aktiv:
Hohe Effizienz





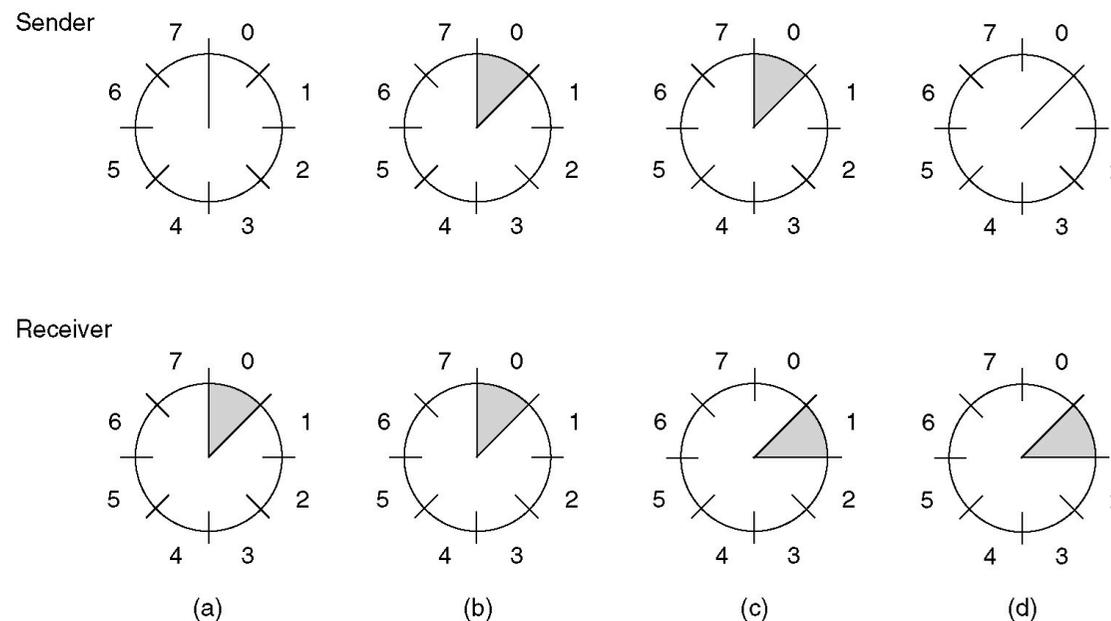
Gleitende Fenster

- **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
- **“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



Beispiel

- **“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



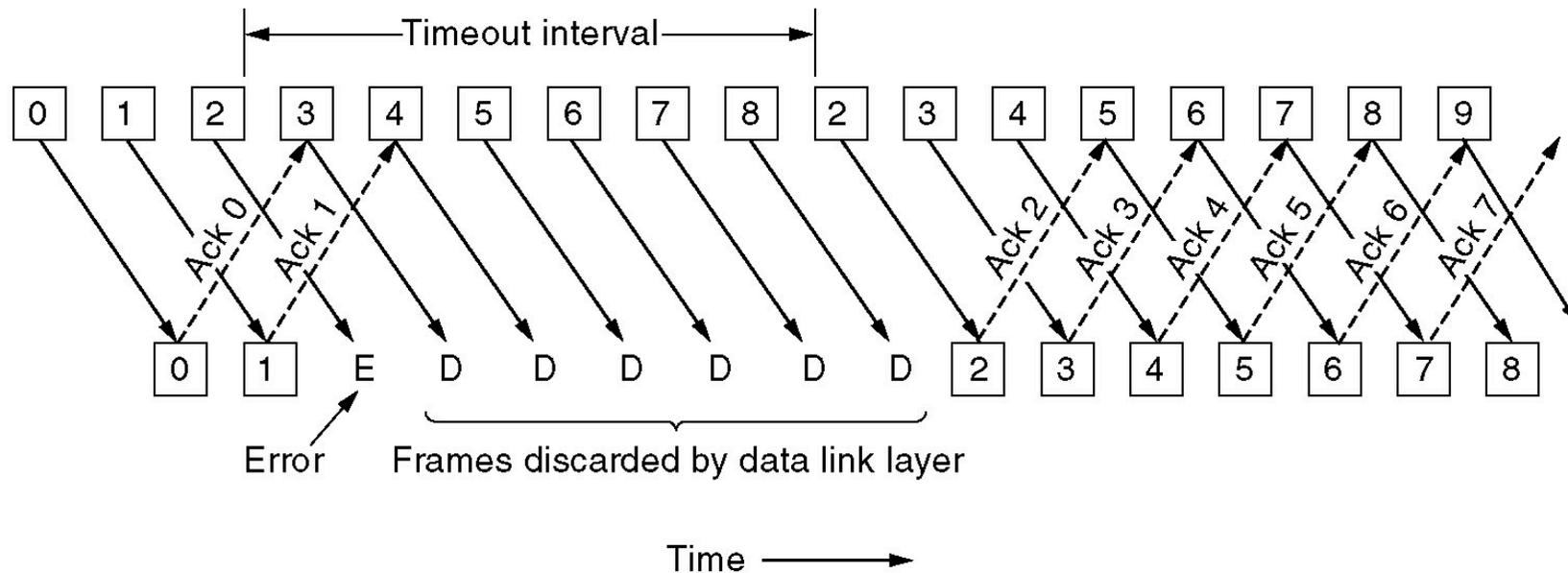
Übertragungsfehler und Empfangsfenster

➤ 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





Go-back-N

- **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 des 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

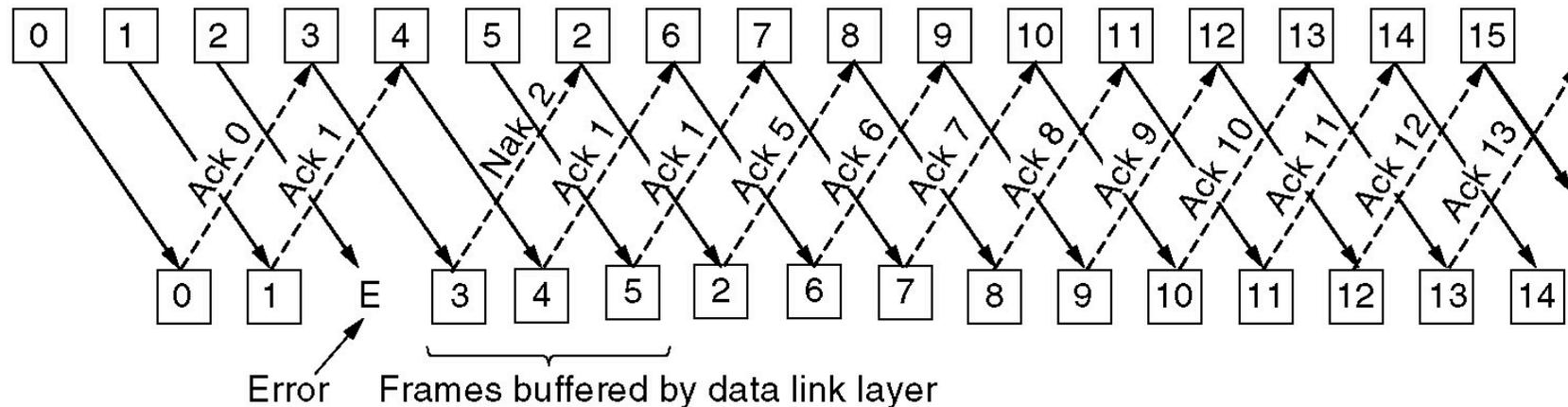


Selektierte Wiederholung

➤ 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



Duplex-Betrieb und Huckepack

➤ 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



Systeme II
Kapitel 4
Mediumzugriff in der
Sicherungsschicht



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer



Der Mediumzugriff in der Sicherungsschicht

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

- **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



Der Mediumzugriff in der Sicherheitsschicht

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

➤ **Statisches Multiplexen**

➤ **Dynamische Kanalbelegung**

- Kollisionsbasierte Protokolle
- Kollisionsfreie Protokolle (contention-free)
- Protokolle mit beschränktem Wettbewerb (limited contention)

➤ **Fallbeispiel: Ethernet**



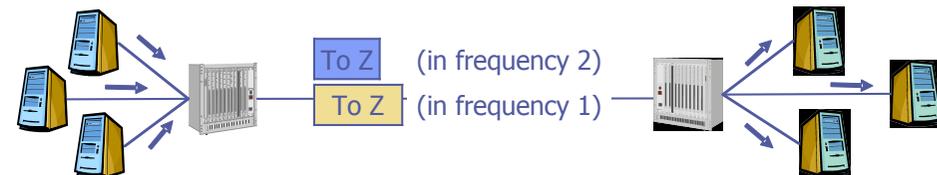
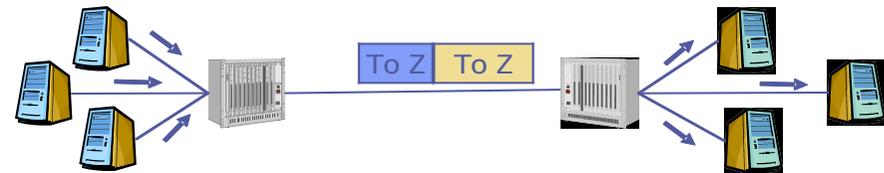
Statisches Multiplexen

➤ **Gegeben sei eine einzelne Leitung (Ressource)**

- Mehreren Kommunikationsverbindungen werden feste Zeiträume/Kanäle (slots/channels) zugewiesen
- Oder: Feste Frequenzbänder werden ihnen zugewiesen ... oder ...

➤ **Gute Lösung falls**

- Feste Datenraten und entsprechenden Anteilen am Kanal
- Quellen lasten die Leitung aus



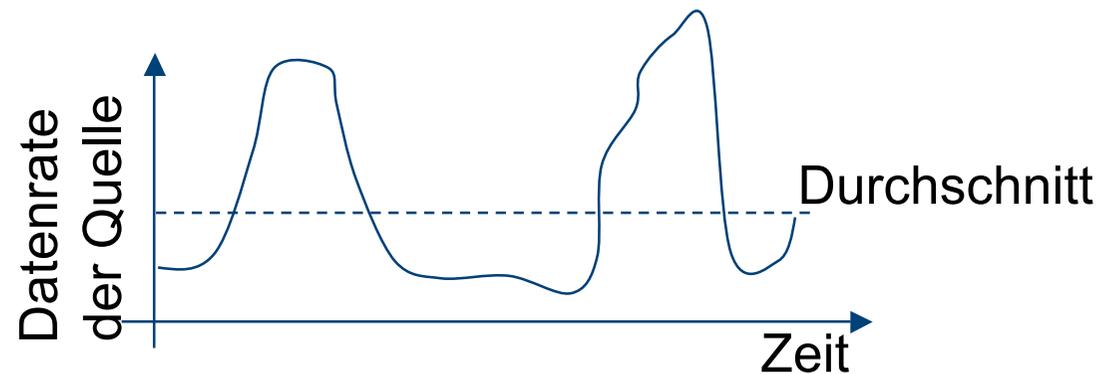


Verkehrsspitzen (*bursty traffic*)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

➤ **Problem: Verkehrsspitzen (*bursty traffic*)**

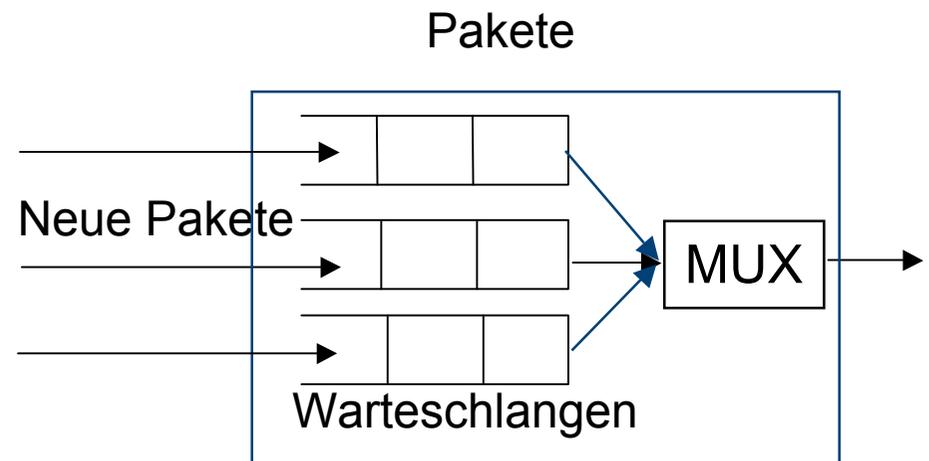
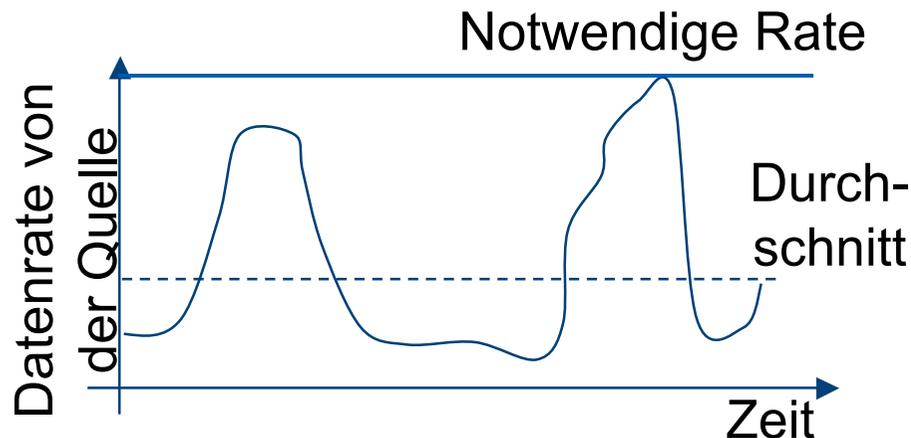
- Definition: Großer Unterschied zwischen Spitze und Durchschnitt
- In Rechnernetzwerken: Spitze/Durchschnitt = 1000/1 nicht ungewöhnlich





Verkehrsspitzen und statisches Multiplexen

- **Leitung für statisches Multiplexen: entweder ... oder**
 - **Genügend große Kapazität um mit dem Peak fertig zu werden**
 - **Verschwendung, da die Durchschnittsrate den Kanal nicht auslasten wird**
 - **Ausgelegt für Durchschnittsrate**
 - **Versehen mit Warteschlangen (queue)**
 - **Vergrößerung der Verzögerung (delay) der Pakete**





Verkehrsspitzen und statisches Multiplexen - Verzögerung

➤ Vergleich der Verzögerung

➤ Ausgangsfall:

- Kein Multiplexing
- Einfacher Datenquelle mit Durchschnittsrate ρ (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



Der Mediumzugriff in der Sicherheitsschicht

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

➤ Statisches Multiplexen

➤ Dynamische Kanalbelegung

- Kollisionsbasierte Protokolle
- Kollisionsfreie Protokolle (contention-free)
- Protokolle mit beschränktem Wettbewerb (limited contention)

➤ Fallbeispiel: Ethernet



Dynamische Kanalzuweisung – MAC

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelbauer

- **Statisches Multiplexing ist nicht geeignet für Datenverbindung mit Spitzen**

- **Alternative: Zuweisung des Slots/Kanals an die Verbindung mit dem größten Bedarf**
 - **Dynamische Medium-Belegung**
 - statt fester

- **Der Mediumzugriff wird organisiert:**
 - ***Mediumszugriff-Protokoll (Medium Access Control protocol - MAC)***



Annahmen

➤ Stationsmodell (terminal model)

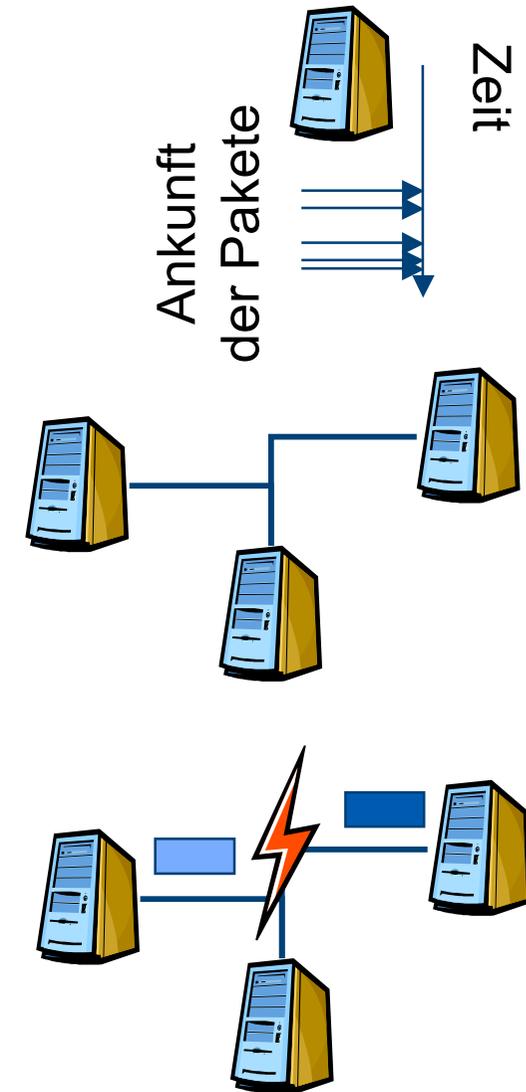
- N unabhängige Stationen möchten eine Leitung/Ressource teilen
- Mögliches Lastmodell:
 - Wahrscheinlichkeit, dass ein Paket im Intervall der Länge Δt erzeugt wird ist $\lambda \Delta t$ für eine Konstante λ

➤ Eine Leitung/Kanal

- für alle Stationen
- Keine weitere Verbindungen möglich

➤ Collision assumption

- Nur ein einfacher Frame kann auf dem Kanal übertragen werden
- Zwei (oder mehr) sich zeitlich überschneidende Frames kollidieren und werden gelöscht
- Noch nicht einmal Teile kommen an





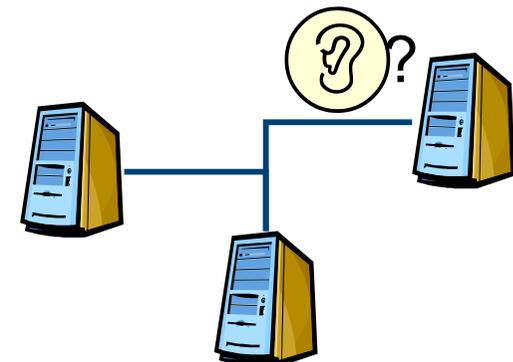
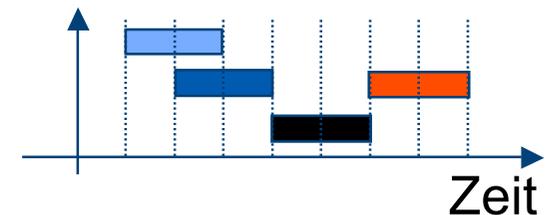
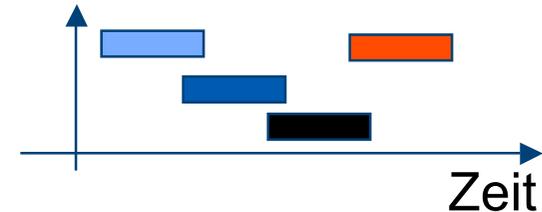
Annahmen

➤ Zeitmodelle

- Kontinuierlich
 - Übertragungen können jeder Zeit beginnen (keine zentrale Uhr)
- Diskret (Slotted time)
 - Die Zeitachse ist in Abschnitte (slots) unterteilt
 - Übertragungen können nur an Abschnittsgrenzen starten
 - Slots können leer (idle), erfolgreich (mit Übertragung) sein oder eine Kollision beinhalten

➤ Träger-Messung (Carrier Sensing)

- Stationen können erkennen ob der Kanal momentan von anderen Stationen verwendet wird
 - Nicht notwendigerweise zuverlässig





Bewertung des Verhaltens

-
- **Methoden zur Bewertung der Effizienz einer Kanaluweisung**

 - **Durchsatz (throughput)**
 - Anzahl Pakete pro Zeiteinheit
 - Besonders bei großer Last wichtig

 - **Verzögerung (delay)**
 - Zeit für den Transport eines Pakets
 - Muss bei geringer Last gut sein

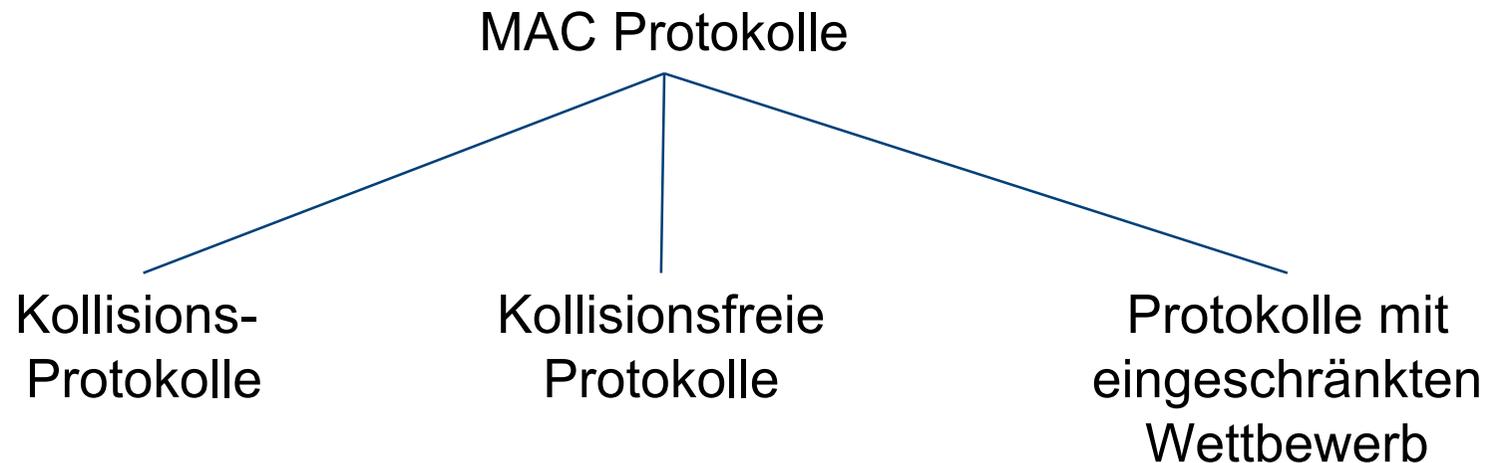
 - **Gerechtigkeit (fairness)**
 - Gleichbehandlung aller Stationen
 - Fairer Anteil am Durchsatz und bei Delay



Mögliche MAC-Protokolle

➤ Unterscheidung: Erlaubt das Protokoll Kollisionen?

- Als Systementscheidung
- Die unbedingte Kollisionsvermeidung kann zu Effizienzeinbußen führen



System mit Kollisionen: **Contention System**



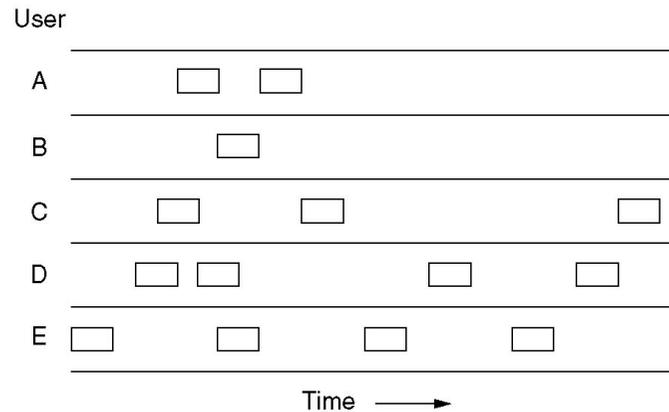
ALOHA

➤ Algorithmus

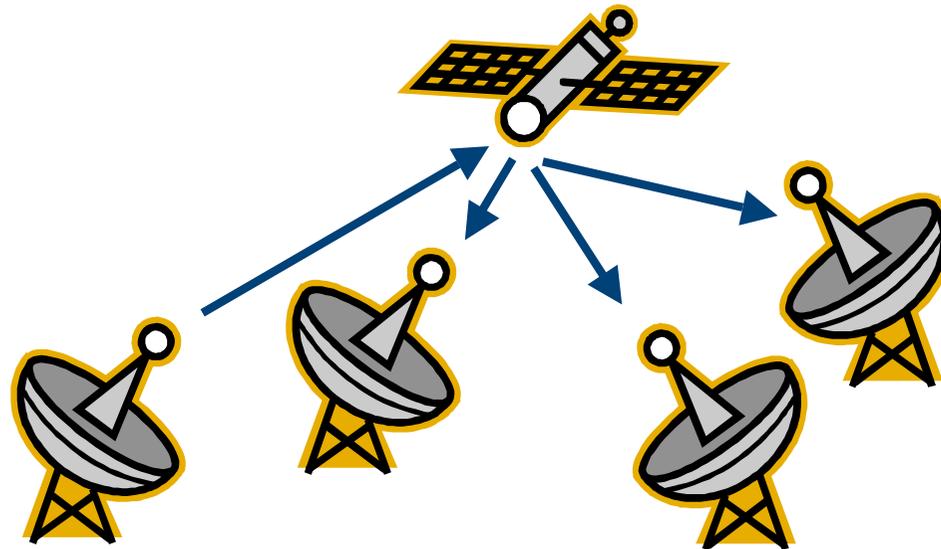
- Sobald ein Paket vorhanden ist, wird es gesendet

➤ Ursprung

- 1985 by Abrahmson et al., University of Hawaii
- Ziel: Verwendung in Satelliten-Verbindung



Pakete werden zu beliebigen Zeiten übertragen





ALOHA – Analyse

➤ Vorteile

- Einfach
- Keine Koordination notwendig

➤ Nachteile

- Kollisionen
 - Sender überprüft den Kanalzustand nicht
- Sender hat keine direkte Methode den Sende-Erfolg zu erfahren
 - Bestätigungen sind notwendig
 - Diese können auch kollidieren



ALOHA – Effizienz

➤ **Betrachte Poisson-Prozess zur Erzeugung von Paketen**

- Entsteht durch “unendlich” viele Stationen, die sich gleich verhalten
- Zeit zwischen zwei Sende-Versuchen ist exponentiell verteilt
- Sei G der Erwartungswert der Übertragungsversuche pro Paketlänge
- Alle Pakete haben gleiche Länge
- Dann gilt

$$P[k \text{ Versuche}] = \frac{G^k}{k!} e^{-G}$$

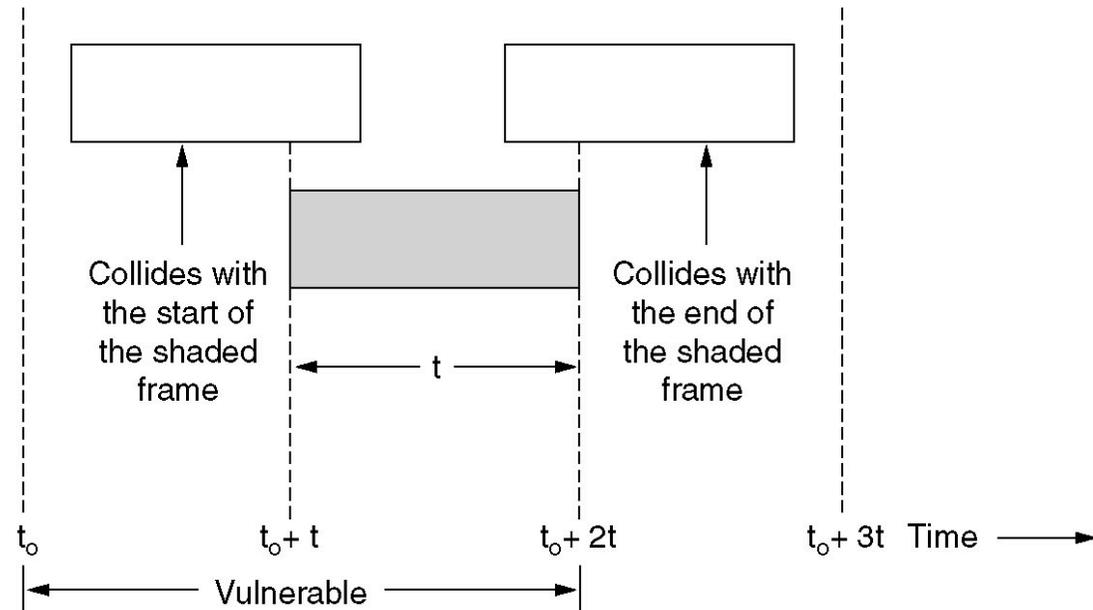
- **Um eine erfolgreiche Übertragung zu erhalten, darf keine Kollision mit einem anderen Paket erfolgen**
- **Wie lautet die Wahrscheinlichkeit für eine solche Übertragung?**



ALOHA – Effizienz

➤ **Ein Paket X wird gestört, wenn**

- ein Paket kurz vor X startet
- wenn ein Paket kurz vor dem Ende von X startet



- **Das Paket wird erfolgreich übertragen, wenn in einem Zeitraum von zwei Paketen kein (anderes) Paket übertragen wird**



Slotted ALOHA

➤ **ALOHA's Problem:**

- Lange Verwundbarkeit eines Pakets

➤ **Reduktion durch Verwendung von Zeitscheiben (Slots)**

- Synchronisation wird vorausgesetzt

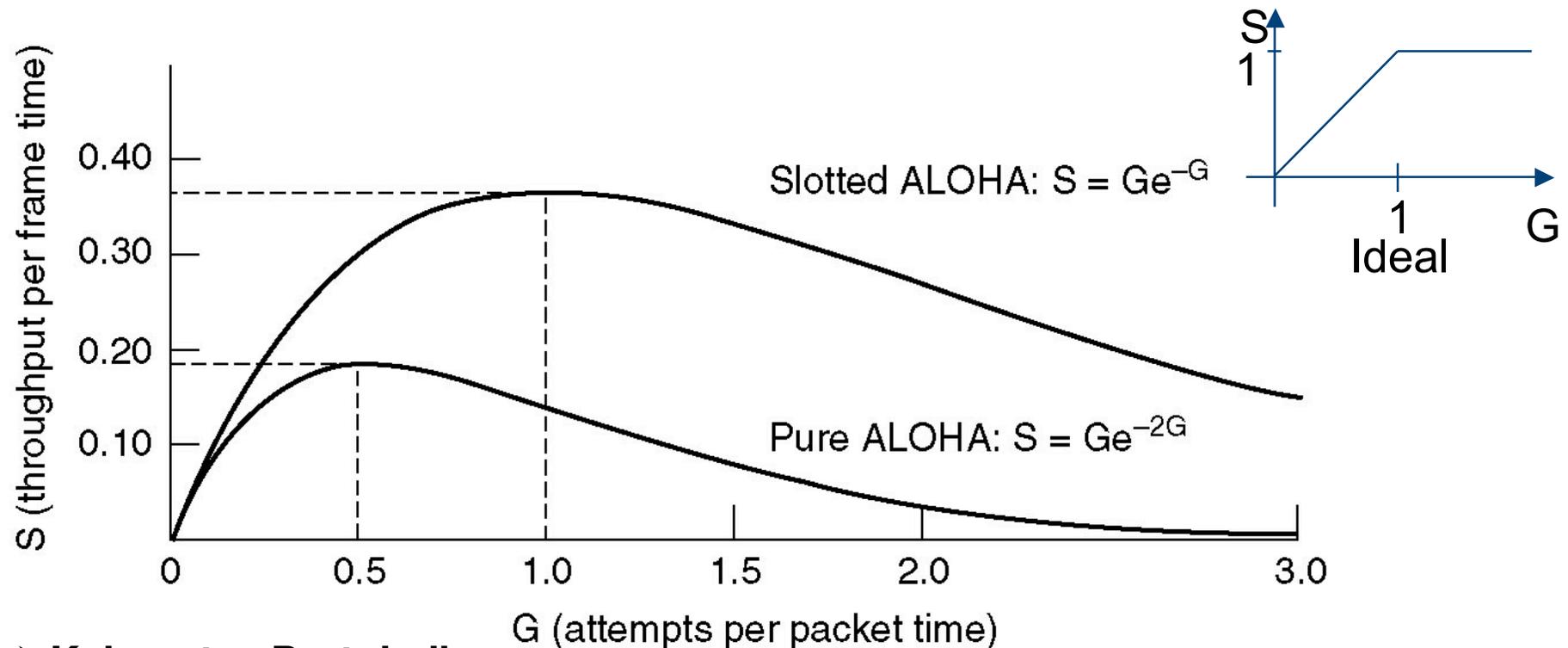
➤ **Ergebnis:**

- Verwundbarkeit wird halbiert
- Durchsatz wird verdoppelt
 - $S(G) = Ge^{-G}$
 - Optimal für $G=1$, $S=1/e$



Durchsatz in Abhängigkeit der Last

- Für (slotted) ALOHA gibt es geschlossene Darstellung in Abhängigkeit von G möglich



- **Kein gutes Protokoll**
 - Durchsatz bricht zusammen, wenn die Last zunimmt

Ende der 6. Vorlesungswoche



Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Systeme II
Christian Schindelhauer
schindel@informatik.uni-freiburg.de