Systeme II

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

Christian Schindelhauer

Sommersemester 2006

7. Vorlesung

17.04.2006

schindel@informatik.uni-freiburg.de

Die Sicherungsschicht

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

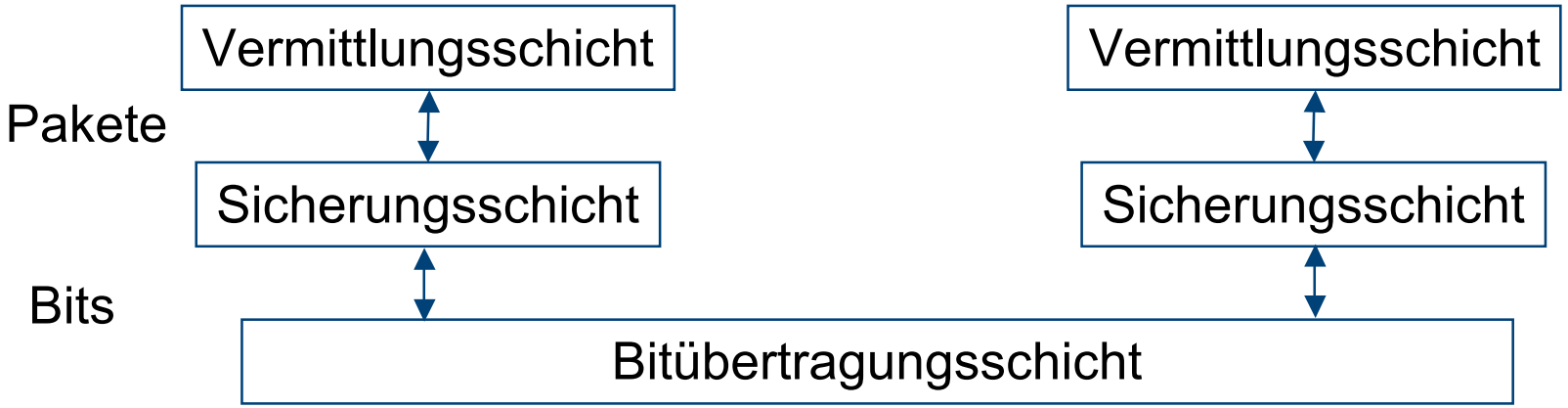
- **Aufgaben der Sicherungsschicht**
  - Dienste für die Vermittlungsschicht
  - Frames
  - Fehlerkontrolle
  - Flusskontrolle
- **Fehlererkennung und Korrektur**
  - Fehlerkorrigierende Kodierungen
  - Fehlererkennende Kodierungen
- **Elementare Sicherungsprotokolle**
  - Simplex
  - „Stop-and-Wait“
  - „Noisy Channel“
- **„Sliding Window“**
  - „1-Bit-Sliding Window“
  - „Go Back N“
  - „Selective Repeat“
- **Protokollverifikation**
  - Endliche Automaten
  - Petrinetze
- **Beispiele**
  - HDLC
  - Internet (PPP)

Einige Folien aus diesem Kapitel
sind aus der Vorlesung „Computer Networks“
von Holger Karl (Universität Paderborn)
übersetzt und „entliehen“ worden

Dienste der Sicherungsschicht

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

- **Situation der Sicherungsschicht**
  - Die Bitübertragungsschicht überträgt Bits
  - Aber unstrukturiert und möglicherweise fehlerbehaftet
- **Die Vermittlungsschicht erwartet von der Sicherungsschicht**
  - Fehlerfreie Übermittlung
  - Übermittlung von struktuiereten Daten
    - Datenpaketen oder Datenströmen
  - Störungslosen Datenfluss



Mögliche Dienste der Sicherungsschicht

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

- **Verlässlicher Dienst?**
  - Das auszuliefernde und das empfangene Paket müssen identisch sein
  - Alle Pakete sollen (irgendwann) ankommen
  - Pakete sollen in der richtigen Reihenfolge ankommen
  - **Fehlerkontrolle** ist möglicherweise notwendig
- **Verbindungsorientert?**
  - Ist die Punkt-zu-Punktverbindung in einem größerem Kontext?
  - Reservierung der Verbindung notwendig?
- **Pakete oder Datenströme (Bitströme)?**

Unterscheidung: Dienst und Implementation

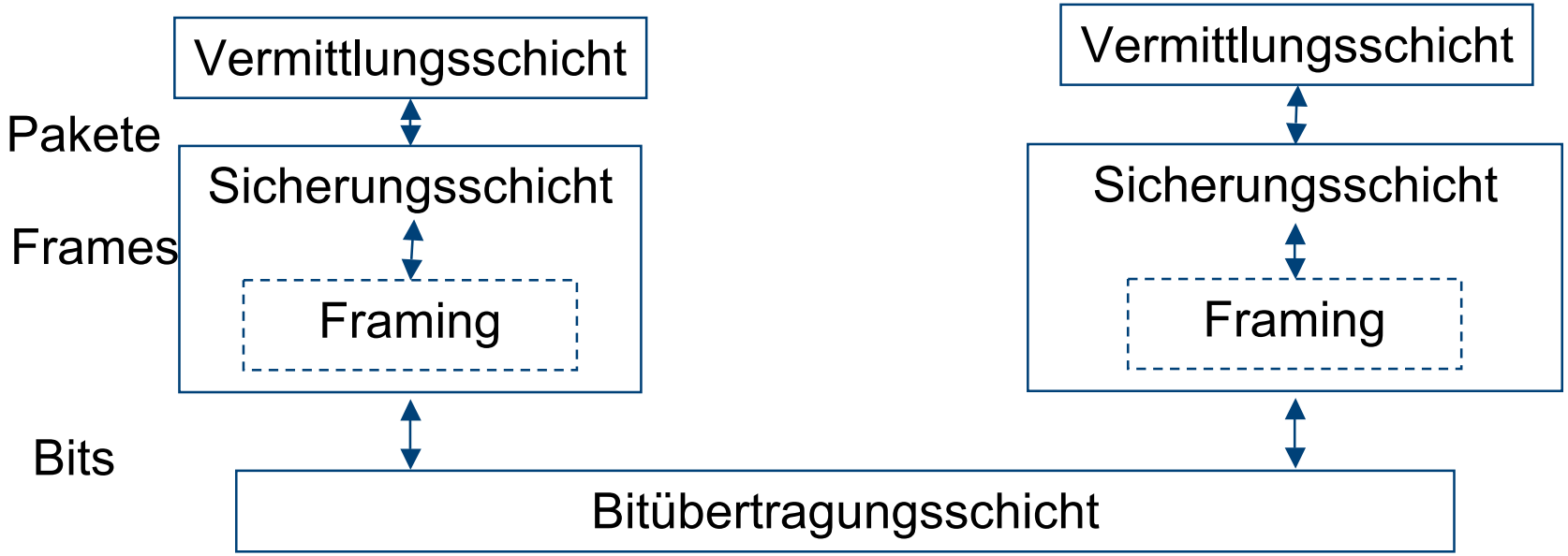
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Beispiel**
  - Verbindungsloser und verlässlicher Dienst wird durch die Vermittlungsschicht gefordert
  - Sicherungsschicht verwendet **intern** verbindungsorientierten Dienst mit Fehlerkontrolle

- **Andere Kombinationen sind möglich**

Frames

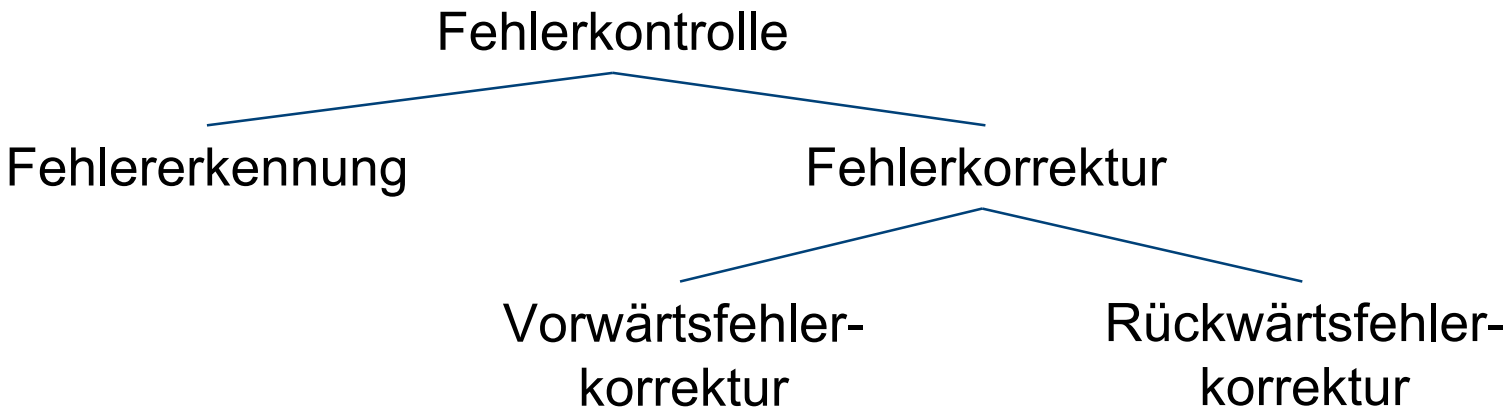
- **Die Bitstrom der Bitübertragungsschicht wird in kleinere "Frames" unterteilt**
  - Notwendig zur Fehlerkontrolle
  - Frames sind Pakete der Sicherungsschicht
- **Frame-Unterteilung (Fragmentierung) und Defragmentierung sind notwendig**
  - Falls die Pakete der Vermittlungsschicht größer sind als die Frames



Fehlerkontrolle

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

- **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
  - Rückwärtsfehlerkorretur (Backward Error Correction)
    - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben



Verbindungsaufbau

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

- **Nutzen von Verbindungen**
  - Kontrolle der Verbindungsstatus
    - Korrektheit des Protokolls
  - Fehlerkontrolle
    - Verschiedene Fehlerkontrollverfahren vertrauen auf gemeinsamen Kontext von Sender und Empfänger
- **Aufbau und Terminierung von Verbindungen**
  - “Virtuelle Verbindungen”
    - Es werden keine Schalter umgelegt
    - Interpretation des Bit-Stroms
  - Durch Frames
  - Besonders wichtig bei drahtlosen Medien
- **Das Problem wird im Rahmen der Transportschicht ausführlich diskutiert**
  - Vgl. Sitzungsschicht vom OSI-Modell

Flusskontrolle

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

- **Problem: Schneller Sender und langsamer Empfänger**
  - Der Sender lässt den Empfangspuffer des Empfängers überlaufen
  - Übertragungsbandweite wird durch sinnlosen Mehrfachversand (nach Fehlerkontrolle) verschwendet

Langsamer Empfänger

Schneller Sender

- **Anpassung der Frame-Sende-Rate an dem Empfänger notwendig**

Frames

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

- **Wo fängt der Frame an und wo hört er auf?**

Übertragener Bitstrom 01100101011101011100101000101010101010101100010

Frame-Anfang?

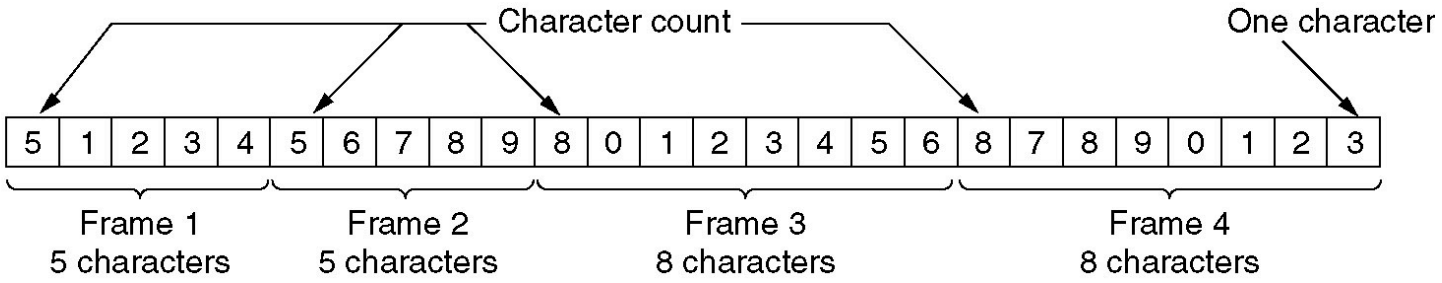
Frame-Ende?

- **Achtung:**
  - Die Bitübertragungsschicht kann auch Bits liefern, wenn der Sender tatsächlich nichts sendet
  - Der Empfänger
    - könnte das Rauschen auf dem Medium interpretieren
    - könnte die Folge 00000000.... liefern
      - Daten oder Kontrollinformation?

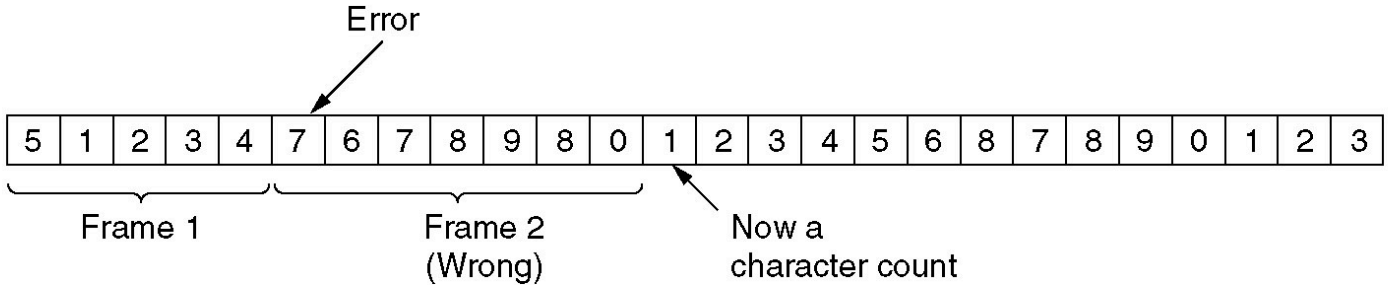
Frame-Grenzen durch Paketlängen?

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

- **Idee: Ankündigung der Bitanzahl im Frame-Header**



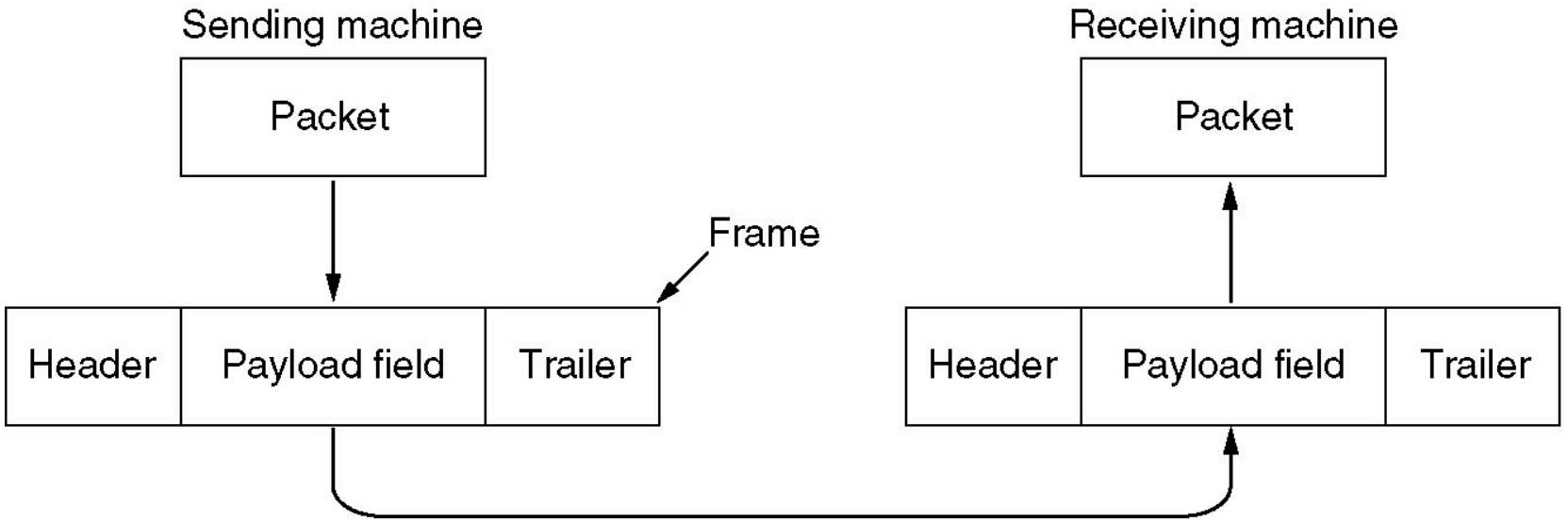
- **Problem: Was, wenn die Frame-Länge fehlerhaft übertragen wird?**
  - Der Empfänger kommt aus dem Takt und interpretiert neue, sinnlose Frames
- **Variable Frame-Größen mit Längeninformation sind daher kein gutes Konzept**



Header und Trailer

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

- **Header und Trailer**
  - Zumeist verwendet man **Header** am Anfang des Frames, mitunter auch **Trailer** am Ende des Frames
  - signalisieren den Frame-Beginn und Frame-Ende
  - tragen Kontrollinformationen
    - z.B. Sender, Empfänger, Frametypen, Fehlerkontrollinformation

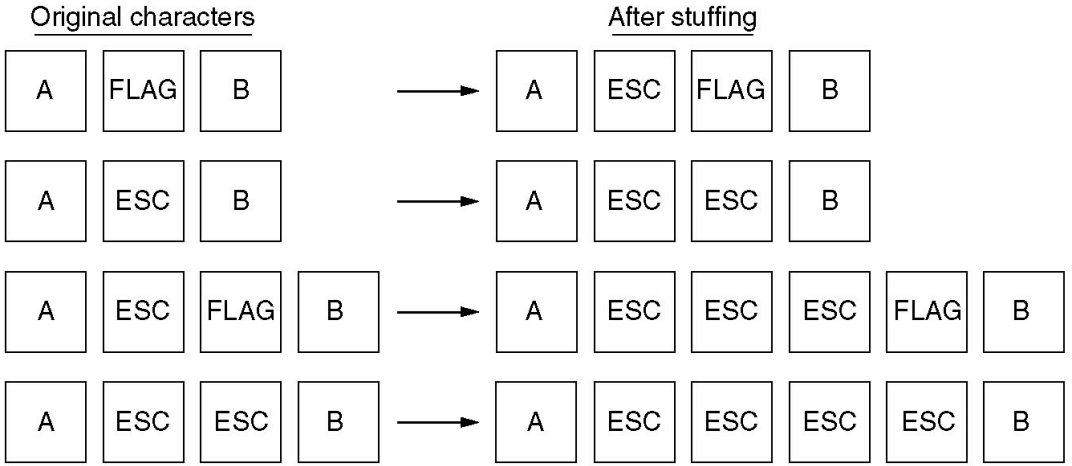


Flag Bytes und Bytestopfen

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

- **Besondere "Flag Bytes" markieren Anfang und Ende eines Frames**

- **Falls diese Marker in den Nutzdaten vorkommen**
  - Als Nutzdatenbyte mit Sonderzeichen (Escape) markieren
    - Bytestopfen (byte stuffing)
  - Falls Sonderzeichen und "Flag-Byte" erscheinen, dito,
    - etc. ,etc.



Frames durch Bit-Sequenzen/Bitstopfen

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

- **Bytestopfen verwendet das Byte als elementare Einheit**
  - Das Verfahren funktioniert aber auch auf Bitebene
- **Flag Bits und Bitstopfen (bit stuffing)**
  - Statt *flag byte* wird eine Bit-Folge verwendet
    - z.B.: 01111110
  - Bitstopfen
    - Wenn der Sender eine Folge von fünf 1er senden möchte, wird automatisch eine 0 in den Bitstrom eingefügt
      - Außer bei den Flag Bits
- **Der Empfänger entfernt eine 0 nach fünf 1ern**

Originale Nutzdate

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Nachdem Bitstopfen

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

Nach der “Entstopfung”

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Frames durch Code-Verletzung

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

- **Möglicher Spielraum bei Bitübertragungsschicht bei der Kodierung von Bits auf Signale**
  - Nicht alle möglichen Kombination werden zur Kodierung verwendet
  - Zum Beispiel: Manchester-Kodierung hat nur tief/hoch und hoch/tief–Übergang

- **Durch "Verletzung" der Kodierungsregeln kann man Start und Ende des Rahmens signalisieren**
  - Beispiel: Manchester – Hinzunahme von hoch/hoch oder tief/tief
    - Selbsttaktung von Manchester gefährdet?

- **Einfache und robuste Methode**
  - z.B. verwendet in Ethernet
  - Kosten? Effiziente Verwendung der Bandbreite?

Fehlerkontrolle

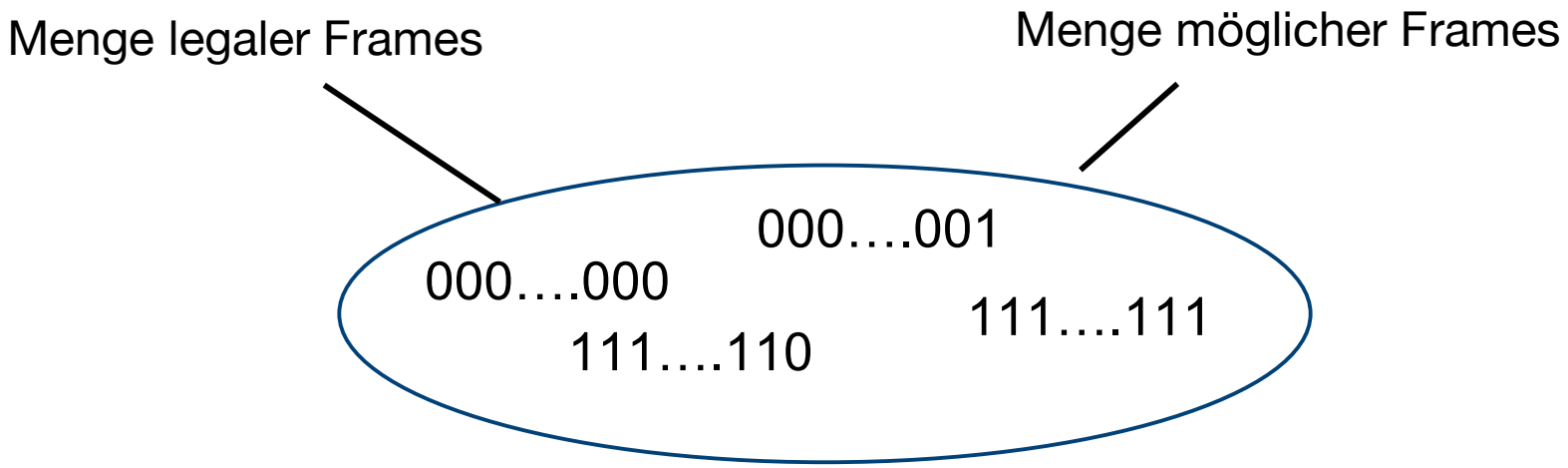
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Aufgaben**
  - Erkennung von Fehlern (fehlerhafte Bits) in einem Frame
  - Korrektur von Fehler in einem Frame
- **Jede Kombination dieser Aufgaben kommt vor**
  - Erkennung ohne Korrektur
    - Löschen eines Frames ohne weiter Benachrichtigung (drop a frame)
    - Höhere Schichten müssen sich um das Problem kümmern
  - Korrektur ohne Erkennung
    - Es werden bestmöglich Bitfehler beseitigt, möglicherweise sind aber noch Fehler vorhanden
    - Sinnvoll, falls Anwendung Fehler tolerieren kann
      - Beispiel: Tonübertragung
    - Prinzipiell gerechtfertigt, weil immer eine positive Restfehlerwahrscheinlichkeit bleibt

Redundanz

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

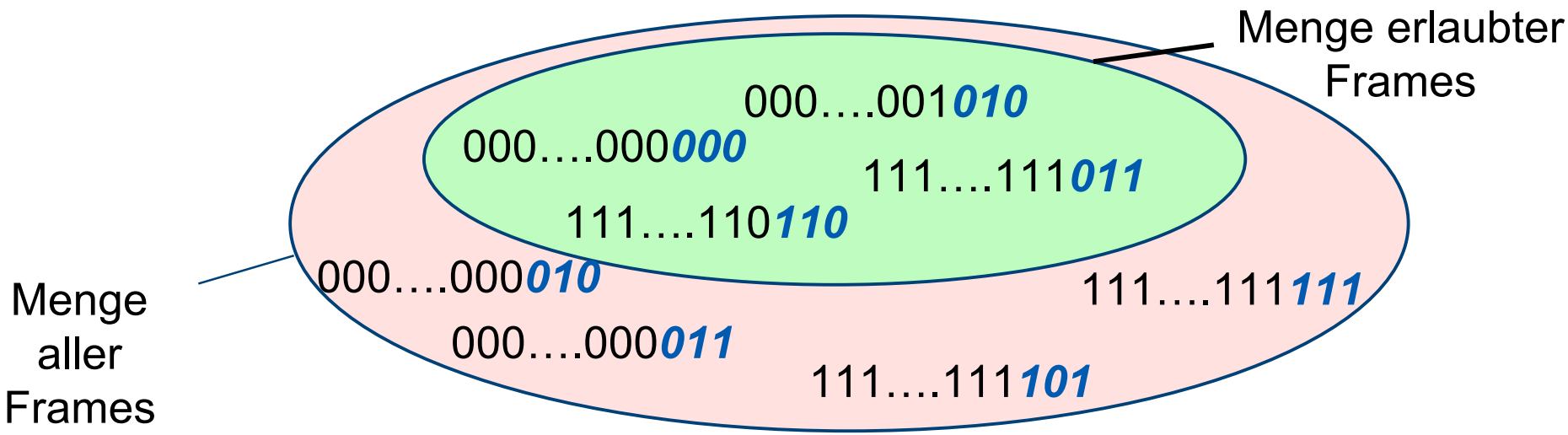
- **Redundanz ist eine Voraussetzung für Fehlerkontrolle**
- **Ohne Redundanz**
  - Ein Frame der Länge m kann 2^m mögliche Daten repräsentieren
  - Jede davon ist erlaubt
- **Ein fehlerhaftes Bit ergibt einen neuen Dateninhalt**



Redundanz

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

- **Kernidee:**
  - Einige der möglichen Nachrichten sind verboten
  - Um dann 2^m legale Frames darzustellen
    - werden mehr als 2^m mögliche Frames benötigt
    - Also werden mehr als m Bits in einem Frame benötigt
  - Der Frame hat also Länge $n > m$
  - *$r=m-n$ sind die* redundanten Bits
    - z.B. Im Header oder Trailer
- **Nur die Einschränkung auf erlaubte und falsche (legal/illegal) Frames ermöglicht die Fehlerkontrolle**



Einfachste Redundanz: Das Paritätsbit

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

- **Eine einfache Regel um ein redundantes Bit zu erzeugen (i.e., n=m+1): Parität**
  - **Odd parity**
    - Eine Eins wird hinzugefügt, so dass die Anzahl der 1er in der Nachricht ungerade wird (ansonsten eine Null)
  - **Even parity**
    - Eine Eins wird hinzugefügt, so dass die Anzahl der 1er in der Nachricht gerade wird (ansonsten wird eine Null hinzugefügt)

- **Example:**
  - Originalnachricht ohne Redundanz: 01101011001
  - Odd parity: 011010110011***1***
  - Even parity: 011010110011***0***

Der Nutzen illegaler Frames

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

- **Der Sender sendet nur legale Frames**
- **In der Bitübertragungsschicht könnten Bits verfälscht werden**
- **Hoffnung:**
  - Legale Frames werden nur in illegale Nachrichten verfälscht
  - Und niemals ein legaler Frame in einem Illegalem
- **Notwendige Annahme**
  - In der Bitübetragungsschicht werden nur eine bestimmte Anzahl von Bits verändern
    - z.B. k bits per frame
  - Die legalen Nachrichten sind verschieden genug, um diese Frame-Fehlerrate zu erkennen

Ende der 7. Vorlesung

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

Systeme II

Christian Schindelhauer

schindel@informatik.uni-freiburg.de

Folien adaptiert aus
Vorlesung *Computer Networks*
von Holger Karl
Universität Paderborn