Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

Computernetze 1

Übung 4

Data Link Layer – Framing und Fehlerbehandlung

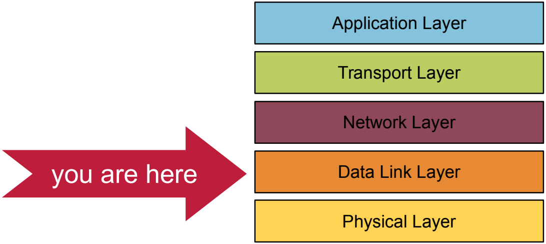
Fynn Schulze, 22. Mai 2025

Technische Universität Braunschweig, IBR

Überblick

- **1) Framing**
- **2) Bitstopfen**
- **3) Hamming-Distanz**
- **4) CRC-Prüfsumme**
- **5) CRC-Prüfsumme**

Überblick


you are here
Application Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

Data Link Layer

Wiederholung: Aufgaben des Physical Layers

Der Physical Layer überträgt einen **Bitstrom**. Der Data Link Layer übergibt einen Bitstrom an den Physical Layer zur Übertragung bzw. bekommt empfangsseitig einen Bitstrom vom Physical Layer angeboten. Der Physical Layer garantiert **nicht**, dass der Bitstrom fehlerfrei ist. Je nach Übertragungsmedium kann sich die Fehlerrate stark unterscheiden
$\Rightarrow$ best-effort service

Aufgaben des Data Link Layers

- Direkte Verbindung zwischen benachbarten Geräten
- Medienzugriff verschiedener Geräte organisieren
- Fehler des Physical Layers erkennen/korrigieren
- Protocol Data Unit (PDU) des Data Link Layers heißt *Frame*

Aufgabe 1: Framing

Der Physical Layer empfängt die Bitfolge A:

A = 0101 1011 1001 0000 1111 1101 1110 0110 1101 1000 1111 0110 1100

Die Bitfolge 11011 ist als Frame Delimiter des Data Link Layers definiert und markiert Beginn oder Ende eines Frames.

a) Erläutern Sie, warum ein Frame Delimiter auf dem Data Link Layer eingesetzt wird.

b) Markieren Sie alle in der Bitfolge A enthaltenen Frames.

1 a) Frame Delimiter

a) Erläutern Sie, warum ein Frame Delimiter auf dem Data Link Layer eingesetzt wird.

Einsatz des Frame Delimiters (FD)

- Bitstrom des Physical Layers muss in Frames eingeteilt werden $\Rightarrow$ FD sind eine mögliche Lösung
- FD ist eine besondere Bitfolge die Start und Ende des Frames markiert
- Physical Layer überträgt möglicherweise auch Bits die zu keinem Frame gehören (Kollision, Rauschen) $\Rightarrow$ Erkennung dieser Bits
- FD darf innerhalb des Frames nicht vorkommen!

Mögliche Alternative: Präambel

- Idee: Erkenne nur den Start eines Frames, Ende siehe Längenfeld

1 b) Markieren der Frames

b) Markieren Sie alle in der Bitfolge A enthaltenen Frames.

Vorgaben

A = 0101 1011 1001 0000 1111 1101 1110 0110 1101 1000 1111 0110 1100

FD = 11011

Lösung

A = 010**1 1011** 1001 0000 1111 **1101 1**110 0**110 11**01 1000 11**11 011**0 1100

Aufgabe 1: Framing

Nehmen Sie an, dass ein Bit des ersten Frame Delimiters fehlerhaft übertragen wurde und folgende Bitfolge B vom Physical Layer empfangen wurde:

B = 0101 **0**011 1001 0000 1111 1101 1110 0110 1101 1000 1111 0110 1100

c) Markieren Sie alle in der Bitfolge B enthaltenen Frames.

d) Erläutern Sie eine Möglichkeit, um die Auswirkungen eines fehlerhaft übertragenen Frame Delimiters abzumildern.

1 c) Markieren der Frames

c) Markieren Sie alle in der Bitfolge B enthaltenen Frames.

Vorgaben

B = 0101 **0**011 1001 0000 1111 1101 1110 0110 1101 1000 1111 0110 1100

FD = 11011

Lösung

B = 0101 **0**011 1001 0000 1111 **1101 11**10 0**110 11**01 1000 11**11 011**0 1100...

Erinnerung: Lösung aus b)

A = 010**1 1011** 1001 0000 1111 **1101 1**110 0**110 11**01 1000 11**11 011**0 1100

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

1 d) Übertragungsfehler abmildern

d) Erläutern Sie eine Möglichkeit, um die Auswirkungen eines fehlerhaft übertragenen Frame Delimiters abzumildern.

Anderer End Frame Delimiter

- Start und Ende eines Frames können unterschieden werden
- Bits außerhalb des Frames können richtig erkannt werden

Auswertung eines vorhandenen Längenfelds

- Fehlerhafter Frame kann als solcher erkannt werden (falsche Länge)
- danach: nächste Bitfolge zwischen FDs als Frame betrachten

Bits zwischen FDs immer als Frame betrachten

- Daten außerhalb von Frames können mit Fehlerprüfung erkannt werden

Aufgabe 2: Bitstopfen

2 a) Geben Sie die Kodierung für die Bitfolge

`1110 1010 0111 1111 1000 1111 0010 1001 1111 1111 1110 0101 1010`

an, wenn ein Bitstopfen nach fünf aufeinander folgenden Einsen verwendet wird.

Lösung

`1110 1010 0111 1111 1000 1111 0010 1001 1111 1111 1110 0101 1010`

`1110 1010 0111 1101 1100 0111 1001 0100 1111 1011 1110 1100 1011 010`

Aufgabe 2: Bitstopfen

2 b) Nennen Sie eine Protokollart bei der Bitstopfen eingesetzt wird und begründen Sie den Einsatz.

Erklärung

- Bitstopfen wird bei *bitorientierten* Protokollen verwendet. Es dient dazu, dass *protokollspezifische* Bitsequenzen (z.B. Flag zur Trennung von Rahmen) nicht im Datenfeld als solche erkannt werden.
- Bitstopfen wird auch zur Synchronisierung eingesetzt, z.B. beim Controller Area Network (CAN).

Hinweis

Manchmal auch „Bit-Füllen“ genannt.
Englisch: „Bit Stuffing“

Aufgabe 3: Hamming-Distanz

Gegeben sei der folgende Code (d.h. eine komplette Liste aller gültigen Codewörter) zur Übertragung von vier verschiedenen Zeichen (A-D):

A	00000
B	10011
C	00110
D	10010

a) Bestimmen Sie die Hamming-Distanz dieses Codes.

b) Bestimmen Sie die Hamming-Distanz des Codes, wenn anstatt D die Codewörter D' = 01010 bzw. D" = 11010 verwendet werden.

c) Begründen Sie, warum die Hamming-Distanz eines Codes als Minimum der Hamming-Distanzen aller gültigen Codewörter definiert ist.

3 a) Hamming-Distanz des Codes

3 a) Bestimmen Sie die Hamming-Distanz dieses Codes.

3 b) Bestimmen Sie die Hamming-Distanz des Codes, wenn anstatt D die Codewörter D' = 01010 bzw. D" = 11010 verwendet werden.

Codewörter

A	00000
B	10011
C	00110
D	10010
D'	01010
D"	11010

Distanzen zwischen den Codewörtern

	A	B	C	D	D'	D"
A		3	2	2		
B			3	1		
C				2		

Lösung 3 a)

Die Hamming-Distanz ist das Minimum aller Distanzen, also **1**.

Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

3 b) Hamming-Distanz mit D’

3 a) Bestimmen Sie die Hamming-Distanz dieses Codes.

3 b) Bestimmen Sie die Hamming-Distanz des Codes, wenn anstatt D die Codewörter D’ = 01010 bzw. D” = 11010 verwendet werden.

Codewörter

A	00000
B	10011
C	00110
D	10010
D’	01010
D”	11010

Distanzen zwischen den Codewörtern

	A	B	C	D	D’	D”
A		3	2	2	2	
B			3	1	3	
C				2	2	

Lösung 3 b) mit D’

Die Hamming-Distanz ist das Minimum aller Distanzen, also **2**.

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

3 b) Hamming-Distanz mit D”

3 a) Bestimmen Sie die Hamming-Distanz dieses Codes.

3 b) Bestimmen Sie die Hamming-Distanz des Codes, wenn anstatt D die Codewörter D’ = 01010 bzw. D” = 11010 verwendet werden.

Codewörter

A	00000
B	10011
C	00110
D	10010
D’	01010
D”	11010

Distanzen zwischen den Codewörtern

	A	B	C	D	D’	D”
A		3	2	2	2	3
B			3	1	3	2
C				2	2	3

Lösung 3 b) mit D”

Die Hamming-Distanz ist das Minimum aller Distanzen, also **2**.

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

3 c) Begründung

3 c) Begründen Sie, warum die Hamming-Distanz eines Codes als Minimum der Hamming-Distanzen aller gültigen Codewörter definiert ist.

Erklärung

Die Hamming Distanz gibt Garantien, wie viele Bitfehler f bei einer Hamming-Distanz von d erkannt bzw. korrigiert werden können.

$$d \geq \mathrm{f} + 1 \,(\text{Erkennung})$$
$$d \geq 2 \cdot \mathrm{f} + 1 \,(\text{Korrektur})$$

Lösung

Diese Garantien müssen auch im 'Worst-Case' gelten, d.h. bei der Unterscheidung der beiden Wörter mit kleinstem Abstand zueinander.

3 c) Beispiel: Fehlererkennung und -behebung

Anderer Code ($d = 3$):

`A: 000000, B: 111000, C: 010110, D: 111111`

Erkennung: $d \geq \mathrm{f} + 1 \Rightarrow \mathrm{f} \leq 2$
Korrektur: $d \geq 2 \cdot \mathrm{f} + 1 \Rightarrow \mathrm{f} \leq 1$

Wir wollen ein A (`000000`) übertragen, aber es treten Fehler auf:

1-Bit-Fehler: `100000`
$\Rightarrow$ eindeutig zuzuordnen zu `A: 000000`

2-Bit-Fehler: `110000`
$\Rightarrow$ Korrektur nicht möglich. `B` wäre sogar näher dran, als `A`!

3-Bit-Fehler: `111000`
$\Rightarrow$ Korrektur nicht möglich. Erkennung nicht möglich!

Aufgabe 4: CRC-Prüfsumme

Die Nachricht 111001101 soll zur Übertragung mit einem CRC versehen werden. Das Generatorpolynom sei $G(x) = x^5 + x^3 + x + 1$.

a) Berechnen Sie die vom Sender verschickte Nachricht durch Polynomdivision.

b) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers für eine fehlerfrei übertragene Nachricht aus.

c) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers aus, wenn das 10. Bit[1] fehlerhaft übertragen wurde.

[1] Die Zählung beginnt links mit dem ersten Bit.

Cyclic Redundancy Check

Redundanz

Zusätzliche Bits werden an die Nachricht angehangen. Diese haben **keinen** Informationswert, sondern dienen nur der Fehlererkennung.

Anwendungsgebiete

- USB
- Ethernet
- Bluetooth
- ...

→ Gut in Hardware umsetzbar

4 a) Versendete Nachricht

Was ist zu tun?

Generatorpolynom: $G(x) = x^5 + x^3 + x + 1 = 101011$

$grad(G(x)) = 5 \Rightarrow 5\ Redundanzbits$

Nutzdaten: $N(x) = 111001101$

entspricht $x^8 + x^7 + x^6 + x^3 + x^2 + 1$

Polynomdivision: $\frac{N(x) \cdot x^5}{G(x)}$

Rest der Polynomdivision ist die gesuchte Prüfsumme:

`11100110100000:101011= ... Rest ...`

Einschub: (Ein kleines bisschen) Algebra

Polynomdivision

Das im Folgenden gezeigte Verfahren gleicht einer schriftlichen Division. Streng genommen handelt es sich aber um eine Polynomdivision. Insbesondere ist das Verfahren nicht übertragbar, wenn die Koeffizienten nicht aus $\mathbb{F}_2$ stammen.

Addition und Subtraktion

Wir arbeiten auf dem Körper $\mathbb{F}_2$ mit den Elementen $\{0, 1\}$. Dort funktionieren Addition und Subtraktion anders als bei den ganzen Zahlen $\mathbb{Z}$. Tatsächlich sind Addition und Subtraktion auf $\mathbb{F}_2$ das gleiche.

4 a) Versendete Nachricht

- Berechnung erfolgt nach XOR-Schema (exklusives ODER)

Definition: XOR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

- $A = B \rightarrow 0$
- $A \neq B \rightarrow 1$

Beispiel

```
 111001
-101011
  10010
```

4 a) Versendete Nachricht

Polynomdivision

```
 1110011010000:101011= 110... Rest ...
-101011
  100101
 -101011
    11100
   -    0
    111001
   -101011
     100100
    -101011
       11110
      -    0    Folie ist voll...
```

4 a) Versendete Nachricht

Polynomdivision (vollständig)

```
 11100110100000:101011= 110... Rest
...1010
-101011
  100101
 -101011
    11100
   -    0
    111001
   -101011
     100100
    -101011
        11110
       -    0
       111100
      -101011
        101110
       -101011
           1010
```

Lösung

Nutzdaten:

111001101

Prüfsumme:

01010

Versendete Nachricht:

11100110101010

4 a) Versendete Nachricht

Fehlererkennung

Mittels CRC-Codes können Fehler nur erkannt werden. Sie können **nicht** korrigiert werden.

Garantien

Fehler können nicht in jedem Fall erkannt werden. Das Verfahren ist insbesondere für zufällige Fehler geeignet, nicht für absichtliche Manipulation. Die Qualität der Fehlererkennung hängt von der Wahl des Generatorpolynoms ab.

4 b) Fehlerfreie Übertragung

Die Nachricht 111001101 soll zur Übertragung mit einem CRC versehen werden. Das Generatorpolynom sei $G(x) = x^5 + x^3 + x + 1$.

a) Berechnen Sie die vom Sender verschickte Nachricht durch Polynomdivision.

b) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers für eine fehlerfrei übertragene Nachricht aus.

c) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers aus, wenn das 10. Bit[2] fehlerhaft übertragen wurde.

[2] Die Zählung beginnt links mit dem ersten Bit.

4 b) Fehlerfreie Übertragung

Polynomdivision

```
 11100110101010:101011= 110... Rest 0
-101011
  100101
 -101011
    11100
   -    0
    111001
   -101011
     100100
    -101011
       11111
      -    0
       111110
      -101011
        101011
       -101011
             00
            - 0
              0
```

Lösung

- Rest ist 0
- Fehlerfreie Übertragung

4 c) Verfälschtes Bit an Stelle 10

Die Nachricht 111001101 soll zur Übertragung mit einem CRC versehen werden. Das Generatorpolynom sei $G(x) = x^5 + x^3 + x + 1$.

a) Berechnen Sie die vom Sender verschickte Nachricht durch Polynomdivision.

b) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers für eine fehlerfrei übertragene Nachricht aus.

c) Führen Sie die Polynomdivision zur Fehlerüberprüfung auf der Seite des Empfängers aus, wenn das 10. Bit[3] fehlerhaft übertragen wurde.

[3] Die Zählung beginnt links mit dem ersten Bit.

4 c) Verfälschtes Bit an Stelle 10

Polynomdivision

```
 11100110111010:101011= 110... Rest 10000
-101011    ⇑
  100101
 -101011
    11100
   -    0
    111001
   -101011
     100101
    -101011
       11101
      -    0
       111010
      -101011
        100011
       -101011
          10000
         -    0
          10000
```

Lösung

- Rest ungleich 0
- Fehler wurde erkannt

Hinweis

- Fehler kann nicht korrigiert werden!

Aufgabe 5: Zyklische Blocksicherung

In einem unsicheren Übertragungskanal soll die Nachricht

$$N = 1010\ \ 0101\ \ 1010$$

übertragen werden und durch die zyklische Blocksicherung (CRC) abgesichert werden. Das Generatorpolynom sei hierfür:
$G(x) = x^4 + x^3 + x + 1$

a) Berechnen Sie die vom Sender übertragene Nachricht durch Polynomdivision.

Während der Übertragung werden 4 Bits verfälscht. Der Empfänger erhält die Nachricht

$$N_e = 1010\ \ 0101\ \ 1011\ \ 0001$$

b) Führen Sie die Fehlererkennung auf Empfängerseite mittels Polynomdivision durch.

5 a) Übertragene Nachricht

Was ist zu tun?

Generatorpolynom: $G(x) = x^4 + x^3 + x + 1 = 11011$

$grad(G(x)) = 4 \Rightarrow 4$ *Redundanzbits*

Nutzdaten: $N(x) = 101001011010$

entspricht $x^{11} + x^9 + x^6 + x^4 + x^3 + x^1$

Polynomdivision: $\frac{N(x) \cdot x^4}{G(x)}$

Rest der Polynomdivision ist die gesuchte Prüfsumme:

`1010010110100000:11011= ... Rest ...`

5 a) Übertragene Nachricht

Polynomdivision (vollständig)

```
 1010010110100000:11011= ... Rest 1010
-11011
  11111
 -11011
    10001
   -11011
     10101
    -11011
      11100
     -11011
        11110
       -11011
          10100
         -11011
           11110
          -11011
             1010
```

Lösung

Nutzdaten:

101001011010

Prüfsumme:

1010

Übertragene Nachricht $U(x)$:

1010010110101010

5 b) Fehlererkennung

Während der Übertragung werden 4 Bits verfälscht. Der Empfänger erhält die Nachricht

$$N_e = \texttt{1010 0101 1011 0001}$$

b) Führen Sie die Fehlererkennung auf Empfängerseite mittels Polynomdivision durch.

Fehlerprüfung am Empfänger

Der Empfänger führt ebenfalls eine Polynomdivision mit der empfangenen Nachricht und dem Generatorpolynom durch.

Ein Fehler ist aufgetreten, wenn der Rest $\neq 0$ ist.

5 b) Fehlererkenung

Polynomdivision

```
 10100101101100001:11011= ... Rest 0
-11011
  11111
 -11011
    10001
   -11011
     10101
    -11011
      11100
     -11011
        11111
       -11011
          10000
         -11011
           10110
          -11011
             11011
            -11011
                 0
```

Lösung

- Rest ist 0
- **Fehlerfreie Übertragung?!**

Achtung

- Fehler wurde **nicht** erkannt!
- Die 4 Bitfehler liegen so, dass ein neues gültiges Codewort entsteht

Zusammenfassung 1/2

Framing

- Erkennung von Frames im Bitstrom des Physical Layers anhand von besonderen Bitfolgen

Bitstopfen

- Besondere Bitfolgen im Payload eines Frames verhindern

Zusammenfassung 2/2

Hamming-Distanz

- Erkennung von Übertragungsfehlern
- Behebung bei genügend großer Hamming-Distanz möglich

CRC

- Erkennung von Übertragungsfehlern
- Keine Fehlerbehebung
- Erkennung von manchen Fehlern nicht möglich

Nächste Übung:

22. Mai 2025