Universität
Paderborn
Fachgebiet
Rechnernetze

Wintersemester 2019/2029

Rechnernetze

Übung 5: Error detection, ARQ, Relationship FEC, CRC, ARQ

2019-11-21

---

1. **Fehlerdetektion**

   Benutzen Sie eine erweiterte Paritätstechnik, um einen fehlerdetektierenden Code zu entwerfen.

   - Eine Folge von $n = k \cdot l$ Bits wird hierbei zeilenweise in eine Matrix mit k Spalten und l Zeilen geschrieben.
   - In einer zusätzlichen Zeile wird für jede Spalte ein Paritätsbit berechnet. (z.B. even parity).
   - Die Daten werden zeilenweise übertragen.

   (a) Beschreiben Sie, wie der Empfänger eine solche Datenfolge auswertet.

   (b) Geben Sie ein Beispiel an, z.B. für $k = 3$ und $l = 4$.

   (c) Wie verhält sich das Verfahren bei einzelnen Bitfehlern oder bei “burst”-Bitfehlern? Wie lange kann eine durchgehende Folge von Bitfehlern sein, die erkannt wird?

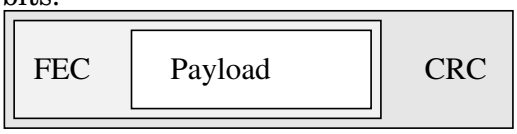
   (d) Wie verändert sich das Verfahren, wenn Sie noch eine zusätzliche Spalte mit zeilenweise berechneten Paritätsbits hinzufügen? Überlegen Sie sich mögliche Vor- und Nachteile.

2. **Fehlererkennung und Fehlerkorrektur**

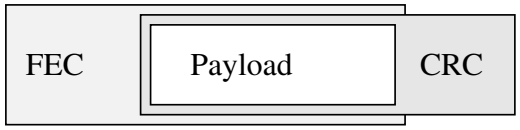
   Das folgende Bild zeigt drei mögliche Szenarien, wie CRC und Vorwärtsfehlerkorrektur (FEC) kombiniert werden können:

In *Szenario 1* berechnet der Sender zunächst die CRC-Prüfsumme zu den Nutzdaten (Payload). Anschließend berechnet er FEC-Bits (Fehlerkorrekturbits) über Nutzdaten- und CRC-Bits.

In *Szenario 2* berechnet der Sender zunächst FEC-Bits über die Nutzdaten. Anschließend berechnet er die CRC-Prüfsumme zu FEC- und Nutzdatenbits.



In *Szenario 3* werden FEC-Bits und CRC-Prüfbits ausschließlich über die Nutzdaten berechnet. Die Reihenfolge, in der die beiden Operationen am Sender ausgeführt werden, ist irrelevant.



Die Konkatenation aus FEC-Bits, Payload und CRC-Bits bildet die zu übertragende Nachricht.

(a) Welche Schritte muss der Empfänger in *Szenario 1* bzw. *Szenario 2* durchführen, nachdem er eine Nachricht empfangen hat?

(b) Sollte der Empfänger in *Szenario 3* zuerst die Fehlerkorrektur (FEC) durchführen oder die CRC-Prüfsumme überprüfen oder ist dies egal?

(c) Es trete ein einzelner Bitfehler **im CRC Teil** der Nachricht auf. In welchen Szenarien führt dies zu einer fehlerhaften CRC Überprüfung und somit zu einem Empfangsabbruch, auch wenn die Fehlerkorrektur (FEC) diesen Fehler korrigieren kann?

(d) Es trete ein einzelner Bitfehler **im Payload Teil** der Nachricht auf. In welchen Szenarien führt dies zu einer fehlerhaften CRC Überprüfung und somit Verwerfen des Paketes, auch wenn die Fehlerkorrektur (FEC) diesen Fehler korrigieren kann?

(e) Warum werden in *Szenario 2* die durch FEC und CRC eingefügten Redundanzbits deutlich schlechter ausgenutzt als in *Szenario 1*?

(f) Eine Übertragung war erfolgreich, wenn der Empfänger nach Auswertung der empfangenen Nachricht den selben Nutzdatenbitstring hat, den der Sender verschickt hat und wenn die CRC-Überprüfung erfolgreich war. Angenommen, es tritt ein einzelner Bitfehler an beliebiger Stelle auf. Ist dann in

Szenario 1 oder in *Szenario 3* die Wahrscheinlichkeit höher, dass die Übertragung erfolgreich war?

3. **Effizienz von ARQ** Wir betrachten einen Link Layer mit folgenden Annahmen:

   - Es werden Pakete der Länge l_P Bits und Acknowledgements der Länge l_A Bits verwendet.
   - Die Ausbreitungsverzögerung sei τ (in beide Richtungen gleich).
   - Die physikalische Datenrate in beiden Richtungen ist r bit/Sekunde.
   - Die Bitfehlerwahrscheinlichkeit ist in beiden Richtungen p, Bits werden von einander unabhängig beschädigt.
   - Es wird kein Vorwärtsfehlerschutz verwendet.
   - Der Übertragungskanal kann vollduplex.
   - Verarbeitungszeiten in Rechnern sind vernachlässigbar.

   (a) Bestimmen Sie die Effizienz (Anteil der Zeit, in der *neue* Daten gesendet werden) und die mittlere Datenrate bei Verwendung von Alternating Bit ARQ. Treffen Sie geeignete Annahmen für die Wahl des Timeouts.

   (b) Plotten Sie die effektive Datenrate abhängig von der Bitfehlerwahrscheinlichkeit für folgende Werte:

$l_P = 100 \cdot 8$	100 Byte Paketlänge
$l_A = 10 \cdot 8$	10 Byte Ack-Länge
$r = 10\,000\,000$	10 MBit/s
$\tau = 0.1/1000$	0.1ms latenz