

Lösungen Übungsblatt 3: Fehlererkennung mittels CRC und Verbindungsverwaltung

Übung

Aufgabe 1 CRC-Summe – Rechnung

Lösung (Binäre Polynomdivision)

- a) Generatorpolynom vom Grad 5 \rightarrow Multiplizieren mit x^5 : An Nachricht anzuhängende Nullen: 5
 $101100100100000 : 101011 = 1001100101$ (Division beinhaltet nur Rest)
 101011

```

-----
001111
-----
011110
-----
111100
101011
-----
101111
101011
-----
001000
-----
010000
-----
100000
101011
-----
010110
-----
101100
101011
-----

```

Hier am Beispiel
**in der Realität sind Generatorpolynome
 natürlich größer! (z.B. Ethernet CRC-32)**

00111 \rightarrow Prüfsumme!

- b) $110100010 : 101011 = 10110$
 101011

```

-----
011111
-----
111110
101011
-----
101011
101011
-----
000000
-----

```

00000 \rightarrow Nachricht wurde höchstwahrscheinlich korrekt übertragen!

Aufgabe 2:

Bei b) bringt eher die Erhöhung der Übertragungsrate einen sichtbaren Erfolg.

In den anderen Fällen wäre vermutlich eine Verkürzung des Übertragungsweges erfolgreicher.

Aufgabe 3:

a) Offensichtlich handelt es sich um eine Simplex-Übertragung. Damit genügt sowohl für den Auf- als auch für den Abbau der Verbindung ein **Zwei-Wege-Handshake**. Der Handshake für den Verbindungsaufbau ist **nötig für die Sequenznummernvergabe**, der Handshake zum **Abbau** ist nötig, um den **vollständigen Empfang der Pakete sicherzustellen**.

b) Zunächst einmal ist die min. **Zeit für den Verbindungsaufbau** zu kalkulieren – dies sind $2 * t_A$ mit:

$$t_A = 2 * 36.000 \text{ km} / 300.000 \text{ km/s} = 240 \text{ ms}$$

Dazu kommt dann natürlich die **Zeit für die Übertragung** der Daten (inkl. Bestätigung). Dies sind zum einen wiederum $2 * t_A$, zum anderen kommt die Übertragungs- bzw. Serialisierungsverzögerung t_S hinzu:

$$t_S = 100 \text{ kB} / 10 \text{ MBit/s} = 800 \text{ kBit} / 10.000 \text{ kBit/s} = 80 \text{ ms}$$

Schließlich muss noch die Ausbreitungsverzögerung für das **Verbindungsabbau-Paket** berechnet werden. Dies ist nur einmal t_A , weil die zurücklaufende Bestätigung den Empfänger eigentlich nicht mehr wirklich interessiert.

Insgesamt ergibt sich also eine minimale Latenz

$$L = (480 + (480 + 80) + 240) \text{ ms} = 1,28 \text{ s.}$$

c) Für die minimale Latenz braucht man hier lediglich die Zeit für das Versenden der Daten einzukalkulieren, **also $t_S + t_A = 320 \text{ ms}$** . Dies zeigt sicherlich deutlich, welche große Verzögerung bei kleinen Datenpaketen der Verbindungsaufbau hervorruft.

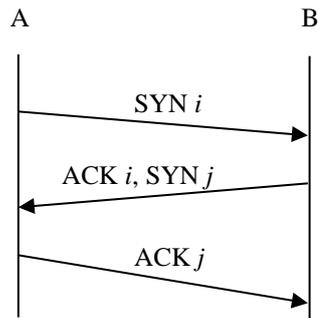
Selbststudium

Aufgabe 3: CRC-Summe – Theorie

Das Generator-Polynom repräsentiert die sechsstellige Binärzahl 101001. Die Berechnung der Prüfsumme ist die Berechnung des Rests bei modulo-2-Division durch diese Zahl. Wäre die Prüfsumme 6 Stellen lang, wäre sie durch den 101001 teilbar (modulo 2!), d.h. es gibt keinen Rest. Eine 5-stellige Prüfsumme ist hingegen nicht durch 101001 teilbar, d.h. es gibt einen Rest. Also benötigt die Prüfsumme 5 Bits.

Aufgabe 4: Verbindungsaufbau – Handshake-Verfahren

Bei einem unzuverlässigen Netz werden typischerweise Sequenznummern für die einzelnen Pakete oder Dateneinheiten vergeben, um so Bestätigungen über den Erhalt der Daten geben zu können. Um dabei Konflikte zwischen alten Verbindungen und der aktuellen Verbindung zu vermeiden, werden diese Sequenznummern beim Aufbau der Verbindung dem jeweiligen Partner mitgeteilt. Beim Drei-Wege-Handshake in beide Richtungen, z. B. bei TCP (SYN bedeutet Verbindungsaufbau, ACK – bestätigung)



Wenn nun eine Instanz selbst etwas senden will, muss sie also die von ihr gewählte Startsequenznummer der Partnerinstanz mitteilen. Da das entsprechende Paket verloren gehen kann, bedarf es einer Bestätigung; kommt das Paket nämlich nicht an, muss es erneut gesendet werden. Dementsprechend ist bei Duplex-Verbindungen ein Drei-Wege-Handshake erforderlich. Bei Simplex-Verbindungen hingegen tut es auch ein Zwei-Wege-Handshake, da B überhaupt keine Sequenznummer zu senden braucht