

## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 1

#### — Aufgaben von Netzwerken —

#### **Musterlösung zu Aufgabe 1:**

- a)
  - Unzuverlässiges Übertragungsmedium: Kanal kann Information verändern, vertauschen, löschen, hinzufügen ...
  - Einzelne Nachrichten müssen als solche erkannt werden (Rahmenbildung, Synchronisation)
  - Zugriff auf das Medium muss geregelt werden (insbesondere bei Broadcast-Medium): Kollisionsvermeidung / -erkennung und Synchronisation
  - Senderate und Empfangsrate müssen abgestimmt sein (Flusskontrolle)
  - Wegewahl, Stauvermeidung, Accounting, Addressierung
  - Fragmentierung und Reassemblierung,
  - Access Token Management, Sitzungs(re-)synchronisation
  - Syntax und Semantik der transportierten Information muss abgestimmt sein
  
- b)
  1. “service”: angebotener Dienst (Was?)
    - z.B. unzuverlässige ende-zu-ende Übertragung
  2. “assumptions about the environment”: Annahmen über die Protokollumgebung (speziell der darunterliegenden Protokollschicht)
    - z.B. Beschaffenheit des Übertragungskanals, Wahrscheinlichkeit bestimmter Fehler, Anzahl Benutzer, ...
  3. “vocabulary of messages”: Menge der erlaubten PDUs (protocol data units)
    - analog zur Menge der erlaubten Wörter einer natürlichen Sprache
    - z.B. “ACK”, “ERR”
  4. “encoding (format)”: Darstellung/Repräsentation von PDUs
    - Bitrepräsentation (Header, Payload, Bedeutung einzelner Bits)
    - Repräsentation als Datenstruktur
  5. “procedure rules”: Verarbeitungsregeln
    - z.B. für bestimmte Nachrichten, Fehler, Timer, etc.
    - Regelmenge muss vollständig und widerspruchsfrei sein

#### **Musterlösung zu Aufgabe 2:**

**HTTP** *Hypertext Transfer Protocol*. Über dieses Protokoll teilt der Web-Browser (Client) dem Server (hier `www.tkn.tu-berlin.de`) mit, welche Seite angezeigt werden soll.

**DNS** *Domain Name System*. Die Beschreibung des Servernamens als Text `www.tkn.tu-berlin.de` ist für Menschen zwar lesbar, für Rechner aber untauglich. Dieses Protokoll ermöglicht die Addressauflösung des Textes in eine Internet (IP) Adresse.

**TCP** *Transmission Control Protocol*. Dieses Protokoll übernimmt die zuverlässige Abwicklung der Anfrage an den Server, d.h. an den Prozess, welcher die Anfrage bearbeitet.

**UDP** *User Datagram Protocol*. Wird im Zusammenhang mit DNS genutzt, um den Namen in eine IP-Adresse aufzulösen (verbindungslose, nicht-zuverlässigen Übertragung).

**IP** *Internet Protocol*. Das Protokoll sorgt dafür, dass ein Paket an den Zielrechner adressiert und weitergeleitet werden kann (ggf. auch Fragmentierung).

**ARP** *Address Resolution Protocol*. IP arbeitet mit anderen Adressen als Ethernet. ARP ermittelt zu einer IP Adresse die passende Hardwareadresse (MAC).

**Ethernet** (IEEE 802.3) Ist nur ein Beispiel für ein LAN (Local Area Network). Alle Rechner, die an ein LAN angeschlossen sind, können direkt erreicht werden (sie sind beispielsweise miteinander mit einem Kabel verbunden).

### Musterlösung zu Aufgabe 3:

- Ein Port ist eine Adresskomponente in Netzwerkprotokollen (TCP, UDP), die zur Zuordnung von Datensegmenten zu bestimmten Diensten verwendet wird. Die Angabe eines Ports erfolgt dabei durch eine Zahl (16-bit unsigned integer).
- Vergabe von Ports erfolgt durch IANA (Internet Assigned Numbers Authority):
  - 0..1023** “well known (privileged) ports”
    - allgemein bekannt, von IANA fest zugeordnete Ports
    - z.B. 80 (HTTP), 53 (DNS), 22 (SSH), 21 (FTP), 123 (NTP)
  - 1024..49152** “registered ports”
    - von Anwendungsherstellern bei IANA registrierte Ports
    - z.B. 5432 (PostgreSQL)
  - 49152..65535** “dynamic (private) ports”
    - variabel einsetzbar, keiner Anwendung zugeordnet

### Musterlösung zu Aufgabe 4:

- a) Problematisch ist das Fehlen einer geeigneten “Abbruchbedingung”: die Nachricht wird ggf. endlos im Netz zirkulieren. Zusätzliche Regel z.B.:

- Merken der weitergeleiteten Nachrichten (Unterdrücken von Duplikaten)
  - Zählerfeld in der Nachricht (TTL) und dekrementieren des Zählers bei Empfang; Weitersenden nur solange  $TTL > 0$
  - Falls Netz synchronisiert ist: absoluter Zeitstempel, Nachricht wird nur bestimmte Zeit weitergereicht
- b) Etwas länger dauert „rumor routing“, auch bekannt als „hot potatoe routing“. Dafür vereinbaren Sie, dass jeder die Anfrage (Antwort) zufällig an einen seiner Nachbarn schickt. Da speziell der Rückweg sehr anfällig ist, können Sie versuchen, Optimierungen einzuführen. Für die Anfrage können Sie „plain flooding“ verwenden und jeder merkt sich den Nachbarn, von dem er die Anfrage zuerst gehört hat, und schickt die Antwort dahin zurück (backward learning).

Die Bezeichnungen für die Verfahren variieren je nach Kontext. Daher immer auf die Beschreibung achten. (Alternativ: „Distance Vector“ oder „Link-State“ + Dijkstra shortest-path, siehe Aufgabenblatt 7)

#### **Musterlösung zu Aufgabe 5:**

Eine Kollision tritt auf sobald mindestens zwei Stationen auf das Medium zugreifen. Die Wahrscheinlichkeit dafür, dass eine Kollision während eines Slots auftritt führt zum Anteil der Slots, die nicht verwendet werden können. Das Ergebnis ist in Abhängigkeit von der Anzahl der Stationen  $n$  und der Wahrscheinlichkeit für einen Zugriff  $p$  anzugeben.

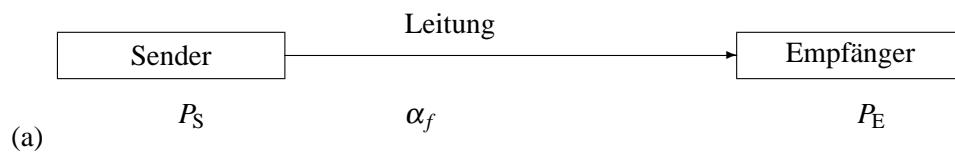
- Wahrscheinlichkeit für keinen Zugriff:  $(1 - p)^n$
- Wahrscheinlichkeit für genau einen Zugriff:  $\binom{n}{1} p (1 - p)^{n-1}$
- Wahrscheinlichkeit für mindestens zwei Zugriffe:  $1 - (1 - p)^n - \binom{n}{1} p (1 - p)^{n-1}$
- Bsp 1:  $n = 2, p = 0.5 \rightarrow 0.25$  bzw. 25%
- Bsp 2:  $n = 8, p = 0.5 \rightarrow 0.96$  bzw. 96%

## Übung zur Vorlesung Kommunikationsnetze

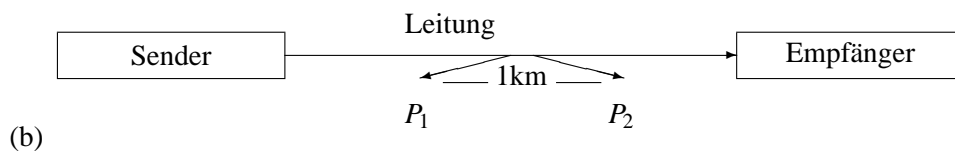
### Aufgabenblatt 2

— Eigenschaften von Kommunikationskanälen —

#### Musterlösung zu Aufgabe 1:



$$\begin{aligned}
 \left( \frac{P_S}{P_E^{(min)}} \right)_{(dB)} &= 10 \cdot \log_{10} \frac{P_S}{P_E^{(min)}} \\
 &= 10 \cdot \log_{10} \frac{P_S}{P_S/1000} \\
 &= 10 \cdot \log_{10} 1000 \\
 &= 30 \text{ dB}
 \end{aligned}$$



$$0.3 \text{ dB} = 10 \cdot \log_{10} \frac{P_1}{P_2} = 10 \cdot \log_{10} \frac{P_1}{(1-x) \cdot P_1} = 10 \cdot \log_{10} \frac{1}{(1-x)}$$

$$0.03 \text{ dB} = -\log_{10} (1-x)$$

$$10^{-0.03} = 1-x$$

$$x = 1 - 10^{-0.03} \approx 0.0667 \Rightarrow 6.7\%$$

(c)

$$l = \frac{30 \text{ dB}}{0.3 \text{ dB/km}} = 100 \text{ km} \quad \text{bzw.}$$

$$(1-x)^l = 0.001 \Rightarrow l \cdot \log_{10}(1-x) = \log_{10}(0.001) \Rightarrow l = \frac{\log_{10} 0.001}{\log_{10}(1-x)} = 100$$

#### Musterlösung zu Aufgabe 2:

(a)  $t_p = \frac{8 \cdot 1000 \text{ Bit}}{100 \cdot 10^6 \text{ Bit/s}} = 80 \cdot 10^{-6} \text{ s} = 80 \mu\text{s}$

- (b) Die Lichtgeschwindigkeit in Glasfaser beträgt ca. 200000 km/s, daher:

$$v = \frac{s}{t_b} \Rightarrow t_b = \frac{100 \text{ km}}{200000 \text{ km/s}} = 5 \cdot 10^{-4} \text{ s} = 500 \mu \text{ s.}$$

- (c)
- $n = \frac{t_b}{t_p} = 6.25 \approx 6$
- Pakete

- (d)
- $t_{\text{ges}} = t_p + t_b + t_b = 1080 \mu \text{ s} \Rightarrow$
- Zeit für ein Paket mit 8000 Bit

$$R = \frac{8000 \text{ Bit}}{1080 \cdot 10^{-6} \text{ s}} \approx 7.41 \cdot 10^6 \text{ Bit/s} = 7.41 \text{ Mbps} \Rightarrow \text{Nettodatenrate}$$

**Musterlösung zu Aufgabe 3:**

- (a)
- Gesucht:**

Pfadverlust in 1 m Entfernung: PL(1 m) [dB]

**Gegeben:**

$$\begin{aligned} \lambda &= c/f && \text{Wellenlänge} \\ &= \frac{300.000 \text{ km/s}}{2,4 \text{ GHz}} \\ &= 0,125 \text{ m} \end{aligned}$$

**Lösung:**

$$\begin{aligned} PL(1 \text{ m}) [dB] &= 10 \cdot \log_{10} \frac{P_t [W]}{P_r(1 \text{ m}) [W]} \\ &= 10 \cdot \log_{10} \frac{P_t}{P_t \cdot \left(\frac{\lambda}{4\pi \cdot 1 \text{ m}}\right)^2} \\ &= 10 \cdot \log_{10} \left(\frac{4\pi \cdot 1 \text{ m}}{0,125 \text{ m}}\right)^2 \\ &= 40.046 \text{ dB} && \approx 40 \text{ dB} \end{aligned}$$

- (b)
- Gesucht:**

minimale Empfangsleistung  $P_{r_{\min}}$

**Gegeben:**

$$\begin{aligned} P_n [dBm] &= -100 \text{ dBm} && \text{Rauschpegel mit Bezugsgröße 1 mW} \\ SNR_{\min} &= 9 \text{ dB} && \text{minimales Signal-Rausch-Verhältnis} \end{aligned}$$

**Definition SNR:**

$$SNR [dB] = 10 \cdot \log_{10} \frac{P_r [W]}{P_n [W]}$$

**Umformung mW  $\Rightarrow$  dBm:**

$$P[\text{dBm}] = 10 \cdot \log_{10} \frac{P[\text{mW}]}{1 \text{ mW}}$$

Im ersten Schritt muss die Rauschleistung von [dBm] in [W] umgerechnet werden:

**Lösung:**

$$\begin{aligned} P_n &= -100 \text{ dBm} \\ -100 \text{ dBm} &= 10 \cdot \log_{10} \frac{P_n [\text{mW}]}{1 \text{ mW}} \\ 10^{-10} &= \frac{P_n [\text{mW}]}{1 \text{ mW}} \\ P_n &= 10^{-13} \text{ W} \end{aligned}$$

Da der minimale SNR bekannt ist, kann die mindestens erforderliche Empfangsleistung berechnet werden:

$$\begin{aligned} \text{SNR}_{\min} &= 9 \text{ dB} \\ 9 \text{ dB} &= 10 \cdot \log_{10} \frac{P_{r_{\min}} [\text{W}]}{P_n [\text{W}]} \\ 10^{\frac{9}{10}} &= \frac{P_{r_{\min}} [\text{W}]}{10^{-13} \text{ W}} \\ P_{r_{\min}} &= 7.943 \cdot 10^{-13} \text{ W} \end{aligned}$$

Ein alternativer Lösungsweg:

$$P_{r_{\min}} = -100 \text{ dBm} + 9 \text{ dB} = -91 \text{ dBm} = 10^{\frac{-91}{10}} \text{ mW} = 7.943 \cdot 10^{-10} \text{ mW}$$

(c) **Gesucht:**

maximale Distanz  $d_{\max}$  zum WLAN AP.

**Gegeben:**

$$\begin{array}{ll} P_t = 30 \text{ mW} & \text{Sendeleistung} \\ P_{r_{\min}} = 7.943 \cdot 10^{-13} \text{ W} & \text{minimale Empfangsleistung} \\ PL(d_0) = 40 \text{ dB} & \text{Referenz-Pfadverlust bei } d_0 = 1 \text{ m} \end{array}$$

Die minimale Empfangsleistung gibt die maximale Distanz vor:

$$P_{r_{\min}} = P_r(d_{\max})$$

Maximal möglicher Pfadverlust:

$$\begin{aligned} PL(d_{max}) &= 10 \cdot \log_{10} \frac{P_t [W]}{P_r(d_{max}) [W]} \\ &= 10 \cdot \log_{10} \frac{0.03 \text{ W}}{7.943 \cdot 10^{-13} \text{ W}} \\ &= 105.77 \text{ dB} \end{aligned}$$

Nach dem Log-distance Path Loss Model gilt:

$$\begin{aligned} PL(d_{max}) [dB] &= PL(d_0) [dB] + 10\alpha \cdot \log_{10} \frac{d_{max}}{d_0} \\ 105.77 \text{ dB} &= 40 + 10 \cdot 4 \log_{10} \frac{d_{max}}{1 \text{ m}} \\ d_{max} &= 10^{\frac{105.77-40}{40}} \text{ m} \\ &= 44.08 \text{ m} \end{aligned}$$

Das Modell sagt eine maximale Entfernung von 44 m voraus. Dieser Wert kann nur eine grobe Abschätzung sein, da im Modell vieles unberücksichtigt bleibt:

- Dämpfung durch individuelle Objekte/Hindernisse
- Multipath-Fading (Überlagerung/Auslöschung durch mehrfaches Empfangen des Signals hervorgerufen bspw. durch Reflexion)
- Interferenz durch andere Nutzer des 2.4 GHz Bandes

(d) **Gegeben:**

$$B = 22 \cdot 10^6 \text{ Hz} \qquad \text{Bandbreite des Kanals}$$

**Gesucht:**

Kanalkapazität  $C$  bei rauschfreiem Kanal

**Lösung:**

3 Bits pro Symbol entsprechen  $2^3$  Signalstufen, d.h.  $M = 8$ .

$$\begin{aligned} C &= 2B \log_2(M) \\ &= 2 \cdot 22 \cdot 10^6 \text{ Hz} \cdot 3 \\ &= 132 \text{ MBit/s} \end{aligned}$$

⇒ Die WLAN Karte erreicht  $\frac{11}{132} \cdot 100 = 8.33\%$  des Wertes.

**Gesucht:**

Kanalkapazität  $C$  bei thermischem Rauschen

**Lösung:**

$$\begin{aligned}C &= B \log_2 \left( 1 + \frac{P_r}{P_n} \right) \\C &= 22 \cdot 10^6 \text{ Hz} \cdot \log_2 \left( 1 + 10^{\frac{9}{10}} \right) \\&= 69.54 \text{ MBit/s}\end{aligned}$$

⇒ Die WLAN Karte erreicht  $\frac{11}{69.54} \cdot 100 = 15.82\%$  des Wertes.

**(e) Lösung:**

$$\begin{aligned}PL(2 \cdot d_{max}) [dB] - PL(d_{max}) [dB] &= 40 + 40 \cdot \log_{10} 2 \cdot d_{max} - (40 + 40 \cdot \log_{10} d_{max}) \\&= 40 + 40 \cdot \log_{10} 2 + 40 \cdot \log_{10} d_{max} - (40 + 40 \cdot \log_{10} d_{max}) \\&= 40 \cdot 0.3 \text{ dB} \\&= 12 \text{ dB}\end{aligned}$$

Bei einer Verdopplung der Distanz, erfährt das Signal eine Dämpfung von 12 dB. Dies kann durch eine Erhöhung der Sendeleistung um einen Faktor  $10^{\frac{12}{10}} = 15.85$  kompensiert werden. Da im 2.4 GHz Band eine maximale Sendeleistung von 100 mW erlaubt ist, wäre damit der zulässige Grenzwert allerdings deutlich überschritten.



## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 3

#### — Warteschlangen —

#### Musterlösung zu Aufgabe 1:

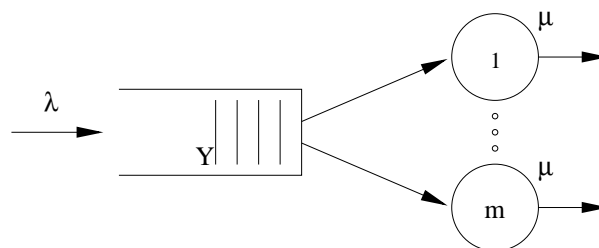


Abbildung 1: Kenngrößen eines Bediensystems

$\lambda$  Mittlere Ankunftsrate

$\mu$  Mittlere Bedienrate

Hier:

$$\lambda = 1000 \frac{\text{Pakete}}{s}$$

$$\mu_I = \frac{40 \text{ Mbit/s}}{10.000 \text{ bit/Paket}} = 4000 \frac{\text{Pakete}}{s}$$

$$\mu_{II} = 2000 \frac{\text{Pakete}}{s}$$

$$m_I = 1$$

$$m_{II} = 2$$

Allgemeine Kendall-Notation: A / B / m / [Y] / [Z]

A: Typ des Ankunftsprozesses (Verteilung der Zwischenankunftszeiten)

B: Typ des Serviceprozesses (Verteilung der Bedienzeiten)

m: Anzahl der Serviceeinheiten

Y: Anzahl der Warteplätze (falls nicht unendlich)

Z: Abfertigungsdisziplin (falls nicht FIFO)

Kurzbezeichnung für Ankunftsprozess und Serviceprozess, z.B.:

M = Exponentialverteilung (Markovian Distribution)

D = Deterministic

G = General

Lösung:

1. M/M/1
2. M/M/1/5
3. M/M/2
4. zweimal M/M/1/5

**Musterlösung zu Aufgabe 2:**

Zustandsübergangsdiagramm für M/M/1 System (Birth-Death-Chain):

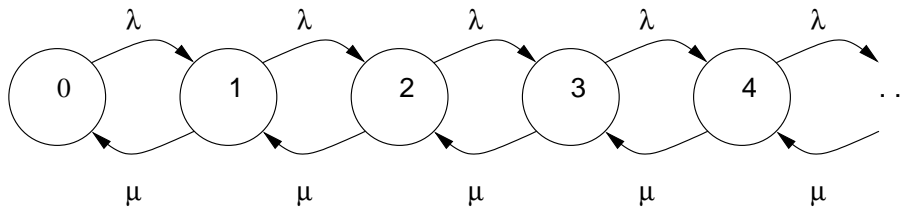


Abbildung 2: Zustandsübergangsdiagramm

In einem stationären M/M/1 System gelten folgende Gleichgewichtsbedingungen:

$$\lambda p_0 = \mu p_1 \Rightarrow p_1 = \frac{\lambda}{\mu} p_0$$

$$\lambda p_1 = \mu p_2 \Rightarrow p_2 = \frac{\lambda}{\mu} p_1 = \left(\frac{\lambda}{\mu}\right)^2 p_0$$

für  $k > 0$  (per Induktion)

$$\lambda p_{k-1} = \mu p_k$$

es folgt

$$p_k = \left(\frac{\lambda}{\mu}\right)^k p_0$$

Es gilt: Summe aller Wahrscheinlichkeiten ist 1

$$\sum_{k=0}^{\infty} p_k = 1$$

Und damit:

$$\begin{aligned} 1 &= p_0 + p_1 + p_2 + \dots = p_0 + \sum_{k=1}^{\infty} p_k \\ &= p_0 + \sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k p_0 \\ &= p_0 \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \\ &\rightarrow \text{Summe einer geometrische Reihe} \\ &= p_0 \frac{1}{1 - \frac{\lambda}{\mu}} \\ \Rightarrow p_0 &= 1 - \frac{\lambda}{\mu} = 1 - \rho \end{aligned}$$

Aus der Einleitung ergeben sich mittlere Ankunfts-, mittlere Bedienrate und damit die Verkehrsdichte wie folgt:

$$\lambda = 1000 \frac{\text{Pakete}}{s}$$

$$\mu = \frac{40 \text{ Mbit/s}}{10.000 \text{ bit/Paket}} = 4000 \frac{\text{Pakete}}{s}$$

$$\rho = \frac{1}{4}$$

Die Wahrscheinlichkeit, dass sich kein Paket im System befindet ist somit:

$$p_0 = 1 - \rho = 0,75$$

Und die Wahrscheinlichkeit für 6 Pakete im System:

$$p_6 = 0,25^6 \cdot 0,75$$

$$= 0,000183105$$

$$\approx 0,02\%$$

$$\Rightarrow \frac{p_6}{p_0} \approx 2,44 \cdot 10^{-4}.$$

### Musterlösung zu Aufgabe 3:

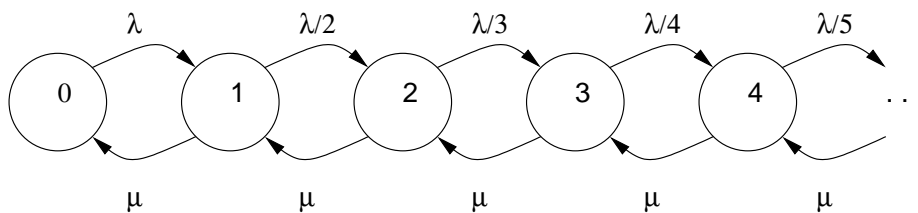


Abbildung 3: Zustandsübergangsdigramm

- (a) Siehe Abbildung 3.
- (b) Beim stationären System kann man mit Hilfe der ersten Definition folgende Gleichungen aufstellen:

$$\lambda p_0 = \mu p_1 \Rightarrow p_1 = \frac{\lambda}{\mu} p_0$$

$$\frac{\lambda}{2} p_1 = \mu p_2 \Rightarrow p_2 = \frac{\lambda^2}{1 \cdot 2 \cdot \mu^2} p_0$$

$$\frac{\lambda}{3} p_2 = \mu p_3 \Rightarrow p_3 = \frac{\lambda^3}{1 \cdot 2 \cdot 3 \cdot \mu^3} p_0$$

$$\vdots$$

Daraus lässt sich

$$p_k = \frac{\lambda^k}{k! \mu^k} p_0$$

Mit  $\rho = \lambda/\mu$

$$= \frac{\rho^k}{k!} p_0$$

induktiv für  $k > 0$  ableiten.

## Beurteilung der Systeme

### Musterlösung zu Aufgabe 4:

$$\begin{aligned}
 E[N] &= \sum_{k=1}^{\infty} k \cdot p_k = \sum_{k=1}^{\infty} k \cdot \rho^k p_0 = \sum_{k=1}^{\infty} k \cdot \rho^k (1-\rho) \\
 &= (1-\rho) \rho \sum_{k=1}^{\infty} k \cdot \rho^{k-1} \\
 &\rightarrow \text{Stammfunktion} \\
 &= (1-\rho) \rho \sum_{k=1}^{\infty} \frac{d}{d\rho} \rho^k \\
 &= (1-\rho) \rho \frac{d}{d\rho} \sum_{k=1}^{\infty} \rho^k \\
 &\rightarrow \text{geometrische Reihe} \\
 &= (1-\rho) \rho \frac{d}{d\rho} \left( \frac{1}{1-\rho} - 1 \right) \\
 &\rightarrow \text{Ableitung} \\
 &= (1-\rho) \rho \frac{1}{(1-\rho)^2} \\
 &= \frac{\rho}{1-\rho}
 \end{aligned}$$

Nach dem Theorem von Little

$$\bar{N} = \lambda \bar{T}$$

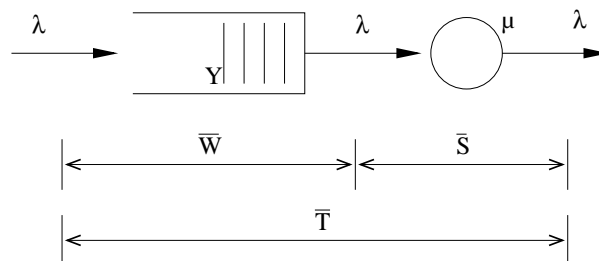


Abbildung 4: Theorem von Little

$$E[N] = \lambda E[T]$$

folgt

$$\begin{aligned}
 E[T] &= \frac{E[N]}{\lambda} \\
 &= \frac{\rho}{1-\rho} \frac{1}{\lambda} \\
 &= \frac{\frac{1}{4}}{1-\frac{1}{4}} \frac{1}{1000} \text{ s} \\
 &= 1/3 \text{ ms}
 \end{aligned}$$

**Musterlösung zu Aufgabe 5:**

Um die drei Alternativen vergleichen zu können, sei  $\rho$  in diesem Fall definiert als  $\rho = \frac{\lambda}{2\mu'}$ . Dann gilt für den Erwartungswert der Anzahl an Aufträgen im System:

1. M/M/1 mit  $\mu = 2\mu'$

$$E[N] = \frac{\frac{\lambda}{\mu}}{1 - \frac{\lambda}{\mu}} = \frac{\frac{\lambda}{2\mu'}}{1 - \frac{\lambda}{2\mu'}} = \frac{\rho}{1 - \rho}$$

(siehe vorherige Aufgabe)

2. M/M/2, Modell als Birth-Death chain gegeben (Abb. 5).

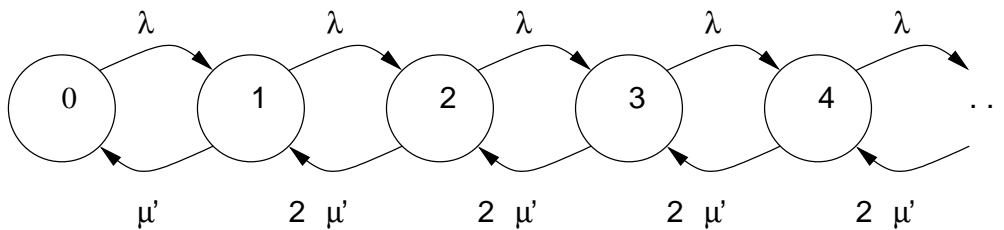


Abbildung 5: M/M/2 System als Birth-Death Chain

Es gilt mit  $\rho = \lambda/(2\mu)$ :

$$p_1 = \frac{\lambda}{\mu'} p_0 = 2\rho \cdot p_0$$

$$p_2 = \frac{\lambda}{2\mu'} p_1 = \rho \cdot p_1 = 2\rho^2 \cdot p_0$$

$$p_3 = \frac{\lambda}{2\mu'} p_2 = \rho \cdot p_2 = \rho^2 \cdot p_1 = 2\rho^3 \cdot p_0$$

⋮

Daraus lässt sich

$$p_k = 2\rho^k \cdot p_0 \quad k > 0$$

$$p_k = \rho^{k-1} \cdot p_1 \quad k > 1$$

induktiv ableiten. Mit  $\sum_{k=0}^{\infty} p_k = 1$  lässt sich dann zeigen:

$$\begin{aligned}
 1 &= p_0 + p_1 + p_2 + p_3 + \dots = \sum_{k=0}^{\infty} p_k \\
 &= p_0 + \rho^0 p_1 + \rho^1 p_1 + \rho^2 p_1 + \dots \\
 &= p_0 + \sum_{k=0}^{\infty} \rho^k p_1 = p_0 + p_1 \sum_{k=0}^{\infty} \rho^k \\
 &\rightarrow \text{geometrische Reihe} \\
 &= p_0 + p_1 \frac{1}{1-\rho} \\
 &= p_0 + 2\rho p_0 \frac{1}{1-\rho} \\
 &= p_0 \left( 1 + \frac{2\rho}{1-\rho} \right) \\
 &= p_0 \frac{1+\rho}{1-\rho} \Rightarrow p_0 = \frac{1-\rho}{1+\rho}
 \end{aligned}$$

Damit folgt:

$$\begin{aligned}
 E[N] &= \sum_{k=1}^{\infty} k \cdot p_k = \sum_{k=1}^{\infty} k 2\rho^k p_0 = \sum_{k=1}^{\infty} k 2\rho^k \frac{1-\rho}{1+\rho} \\
 &= 2\rho \frac{1-\rho}{1+\rho} \sum_{k=1}^{\infty} k \rho^{k-1} \\
 &\rightarrow \text{Stammfunktion} \\
 &= 2\rho \frac{1-\rho}{1+\rho} \sum_{k=1}^{\infty} \frac{d}{d\rho} \rho^k \\
 &\rightarrow \text{konstanter Faktor, geometrische Reihe} \\
 &= 2\rho \frac{1-\rho}{1+\rho} \cdot \frac{d}{d\rho} \left( \frac{1}{1-\rho} - 1 \right) \\
 &\rightarrow \text{Ableitung} \\
 &= 2\rho \frac{1-\rho}{1+\rho} \frac{1}{(1-\rho)^2} \\
 &= \frac{2\rho}{1-\rho^2}
 \end{aligned}$$

3. Ein von zwei M/M/1 Systemen als Birth-Death Chain gegeben (Abb. 6).

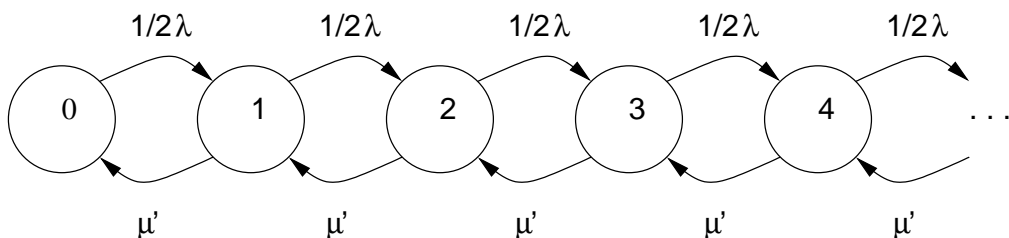


Abbildung 6: Ein von zwei M/M/1 Systemen als Birth-Death Chain

Für die mittlere Anzahl von Aufträgen gilt damit:

$$\frac{\frac{(1/2)\lambda}{\mu'}}{1 - \frac{(1/2)\lambda}{\mu'}} = \frac{\rho}{1-\rho}$$

Für das Gesamtsystem folgt deshalb:

$$E[N] = 2 \frac{\rho}{1 - \rho}$$

In der folgenden Abbildung 7 (siehe nächste Seite) ist für die drei Alternativen die mittlere Anzahl an Aufträgen im System abhängig von der Auslastung  $\rho$  des Systems dargestellt (setze  $\rho = \text{rho}$ ).

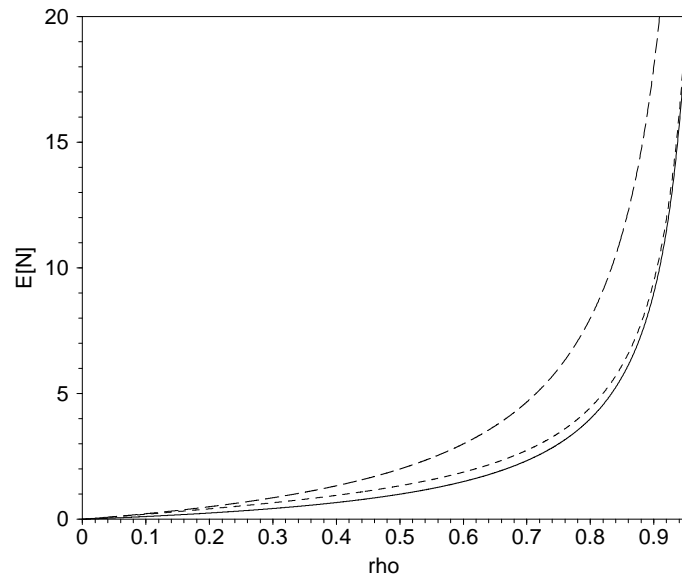


Abbildung 7: Vergleich der drei Alternativen

Die durchgezogene Linie steht für die Alternative 1, die kurz gestrichelte Linie steht für die Alternative 2, und die lang gestrichelte Linie steht für die Alternative 3.

### Nachwort

Die Warteschlangen-Theorie wurde hier an einem etwas konstruierten Beispiel eingeführt. Für die Auslegung von Routern und Switchen ist ihre Anwendung um einiges natürlicher und von praktischer Bedeutung.



## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 4

#### — Erkennung und Korrektur von Bitfehlern —

#### Musterlösung zu Aufgabe 1:

1) Mögliche Verfahren:

– Paritätsbits, Prüfsummen, CRC, Hashes

2) Fehlererkennung findet in der Regel auf Schicht 2 (Sicherheitsschicht, Data-Link Layer) und z.T. auch auf Schicht 4 (Transportschicht, Transport Layer) statt.

#### Musterlösung zu Aufgabe 2:

Beispiel-Bitfehlerraten bekannter Systeme (Unit 7, Folie 3):

- X-25:  $10^{-5}$
- LAN:  $10^{-7}$
- Glasfaser:  $10^{-9}$

(a) Sei  $X$  die Anzahl verfälschter Bits. Zufallsvariable  $X$  ist *binomialverteilt* ( $B_{n,p}$ ), d.h.

$$p_k = \binom{n}{k} p^k (1-p)^{(n-k)} = P(X = k)$$

z.B.

- $n = 8, k = 1 \Rightarrow p_k \approx 7,4\%$
- $n = 8, k = 2 \Rightarrow p_k \approx 0,3\%$
- $n = 8, k = 3 \Rightarrow p_k \approx 0,005\%$

(b) Ein Paket ist fehlerhaft, wenn mindestens ein Bit fehlerhaft ist, d.h.

$$P(X > 0) = 1 - P(X = 0) = 1 - (1-p)^n$$

z.B.

- $n = 8 \Rightarrow P(X > 0) \approx 7,7\%$

- (c) Sei  $Y$  die Anzahl notwendiger Übertragungen.  $Y = k$  bedeutet, daß zunächst  $k - 1$  Pakete fehlerhaft, das  $k$ -te Paket jedoch korrekt übertragen werden. Daraus ergibt sich zunächst:

$$P(Y = k) = P(X > 0)^{(k-1)}P(X = 0)$$

Um die mittlere Anzahl von Wiederholungen zu bestimmen, wird der Erwartungswert der Zufallsvariable  $Y$  betrachtet:

$$\begin{aligned} E[Y] &= \sum_{k=1}^{\infty} kP(Y = k) \\ &= \sum_{k=1}^{\infty} k(1 - (1 - p)^n)^{(k-1)}(1 - p)^n \\ &\rightarrow \text{sei } q = 1 - (1 - p)^n \\ &= \sum_{k=1}^{\infty} kq^{(k-1)}(1 - q) = (1 - q) \sum_{k=1}^{\infty} kq^{(k-1)} \\ &\rightarrow \text{Stammfunktion} \\ &= (1 - q) \sum_{k=1}^{\infty} \left( \frac{d}{dq} q^k \right) = (1 - q) \left( \frac{d}{dq} \sum_{k=1}^{\infty} q^k \right) \\ &\rightarrow \text{geometrische Reihe} \\ &= (1 - q) \left( \frac{d}{dq} \left( \frac{1}{1 - q} - 1 \right) \right) \\ &\rightarrow \text{Ableitung} \\ &= (1 - q) \frac{1}{(1 - q)^2} = \frac{1}{1 - q} \\ &= \frac{1}{(1 - p)^n} \\ &= \frac{1}{P(X = 0)} \end{aligned}$$

Wichtiges Ergebnis! Z.B.

- $n = 8 \Rightarrow E[Y] = 1,0837 \dots$  (1084 Übertragungen für 1000 Pakete)

### Fehlererkennung

Um mit Bitfehlern in Paketen umgehen zu können, müssen diese zunächst erkannt werden. Eine sehr effiziente Möglichkeit dafür stellen Verfahren nach dem Prinzip der zyklischen Redundanzprüfung (engl. *cyclic redundancy check* – **CRC**) dar. Im Folgenden sollen die zu übertragende Bitfolge 1101101 und das Generatorpolynom  $G(x) = x^3 + x + 1$  betrachtet werden.

### Musterlösung zu Aufgabe 3:

Die Theorie der Fehlerkorrektur stützt sich mathematisch auf das Rechnen in Galois-Körpern. Eine hinreichende Einführung kann hier nicht geleistet werden. Glücklicherweise brauchen wir daraus hier nur einen kleinen Ausschnitt. Wir rechnen im  $\text{GF}(2)$ , d.h.

- es gibt nur zwei Zahlen: 1 und 0 (davon ist die 1 das Eins-Element und die 0 das Null-Element) ;-)

- Das inverse Element (-1) existiert
- $1 + 1 = 0$  weil das Ergebnis wieder eine Zahl im GF(2) sein muss. Die Addition im GF(2) kann man auch mit einem speziellen Symbol kennzeichnen:  $\oplus$  – die Addition modulo 2.
- das Kommutativ-, Assoziativ- und Distributivgesetz gelten

Eine Bitfolge lässt sich auf verschiedene Arten beschreiben:

$$\begin{array}{rccccccc} \text{Vektor:} & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \text{Polynom:} & 1x^6 & +1x^5 & +0x^4 & +0x^3 & +1x^2 & +0x^1 & +1x^0 \\ & 1x^6 & +1x^5 & & & +1x^2 & & +1 \end{array}$$

Das ist ganz ähnlich der gewohnten Zahlendarstellung, z.B. lässt sich die Zahl 123 als  $1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$  darstellen.

Das Generator-Polynom ist ein Polynom vom Grad drei und lässt sich kompakt als 1011 darstellen. Im Folgenden rechnen wir mit dieser kompakten Darstellung ähnlich zur schriftlichen Division.

Zunächst hängen wir an die zu übertragende Bitfolge drei Bit an (entspricht dem Grad des Generatorpolynoms) und teilen wie bei einer normalen Polynom Division durch das Generatorpolynom. Eine Bitfolge ist dann teilbar, wenn sie mindestens vom Grad des Generatorpolynoms ist und an der höchsten Stelle eine 1 steht. Wir sind nur am Rest interessiert und schreiben daher das Ergebnis nicht mit.

$$\begin{array}{r} 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ : \ 1 \ 0 \ 1 \ 1 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 1 \ 0 \ 1 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 1 \ 0 \ 0 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 1 \ 1 \ 1 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 0 \ 0 \ 0 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 1 \ 0 \ 0 \\ \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 1 \ 1 \ \leftarrow \text{Rest} \end{array}$$

Statt der drei aufgefüllten Bits wird der Rest übertragen, d.h. es wird 1101101111 übertragen.

#### Musterlösung zu Aufgabe 4:

Der Empfänger führt die gleiche Rechnung durch. Allerdings darf jetzt kein Rest bleiben, sonst war die Übertragung nicht korrekt. Für die erste Bitfolge (1101101101) ergibt sich folgende Rechnung:

$$\begin{array}{r}
 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ : 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 1\ 0\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 1\ 0\ 0 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 1\ 1\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 1\ 0 \leftarrow \text{Rest}
 \end{array}$$

Da jetzt ein Rest bleibt, war die Übertragung **nicht korrekt**. Für die zweite Bitfolge (01011011) sieht die Rechnung wie folgt aus:

$$\begin{array}{r}
 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ : 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 0\ 0 \\
 \\
 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 0\ 0
 \end{array}$$

Die Verfälschung in der Übertragung wird **nicht erkannt!**

### Fehlerkorrektur

In den vorherigen Aufgaben haben Sie Methoden untersucht, mit denen Fehler bei der Übertragung erkannt werden können. In der Vorlesung haben Sie auch Codes kennen gelernt, mit denen man in der Lage ist, Fehler zu korrigieren.

### Musterlösung zu Aufgabe 5:

- (a) Ein Hamming-Code wird meistens mit dem Tupel  $(n,i)$  angegeben. Dabei bedeutet

**n** Gesamtzahl der Bits in einem Block

**i** Davon die Anzahl der Nutz- (Informations-, oder Nachrichten-) Bits

Es existieren 4 Nachrichtenbits, daraus ergeben sich  $2^4 = 16$  gültige Codewörter.

- (b) Mit der Kodiervorschrift erhält man die Kontrollstellen 111, 010, 110 und 101. Die grafische Darstellung der Kodiervorschrift (Abb. 1) ist hilfreich.

- (c)

Fehlerposition :	$f_1$	$f_2$	$m_1$	$f_3$	$m_2$	$m_3$	$m_4$
$s_1$	1	0	1	0	1	0	1
$s_2$	0	1	1	0	0	1	1
$s_3$	0	0	0	1	1	1	1

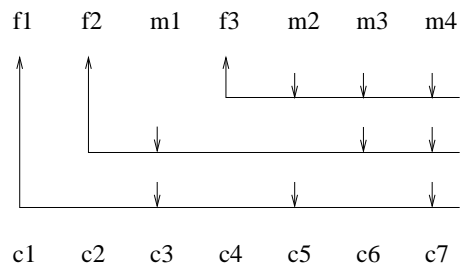


Abbildung 1: Codiervorschrift

- (d) Ein Code mit einer Hamming-Distanz von  $n$  kann bis zu  $n - 1$  Fehler erkennen und  $(n - 1)/2$  Fehler korrigieren. Der (7,4) Hamming-Code kann 2 Bitfehler erkennen (und 1 Bitfehler korrigieren).
- (e)

<i>Fehlerposition : <math>f_1</math> und <math>m_1</math>   <math>m_2</math> und <math>m_3</math></i>		
$s_1$	0	1
$s_2$	1	1
$s_3$	0	0

Der Decoder erkennt einen Fehler an Stelle 2 bzw. 3 statt der richtigen Fehler und würde so die falschen Bits korrigieren. Das stellt keinen Widerspruch zu Aufgabe (d) dar, da nur von der *Erkennung* nicht aber von der *Korrektur* zweier fehlerhaften Bits die Rede war. Es können nur  $c$  Fehler korrigiert werden, wenn  $d_{\min}(C) \geq 2c + 1$  gilt, d.h ein fehlerhaftes Bit.

Entweder muss der Decoder oder der Code erweitert werden. Beim *erweiterten* Hamming-Code wird ein Paritätsbit gesendet. Man kann den Decoder jedoch auch lediglich überprüfen lassen, ob es sich überhaupt um ein gültiges Codewort handelt. Siehe Hamming-Distanz.

## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 5

#### — ARQ-Protokolle —

#### Musterlösung zu Aufgabe 1:

- Rahmenbildung (Framing) – Umsetzung von Datenpaketen auf Bitstrom, Prüfsummen
- Schutz gegen Übertragungsfehler – FEC, ARQ
- Keine Überlastung des Empfängers – Flusskontrolle
- In der Transportschicht: keine Überlastung der Router auf dem Weg: Überlastabwehr – z.B. durch Fenster basierte Verfahren, token bucket etc.

#### Musterlösung zu Aufgabe 2:

(a) Skizze!

$$\text{Kanaleffizienz} = \frac{\text{Generierungszeit der Nachricht}}{\text{Gesamtzeit}} = \eta \in [0, 1]$$

Für Send and Wait gilt:

$$\eta = \frac{t_{\text{Paket}}}{t_{\text{Paket}} + \tau + t_{\text{Ack}} + \tau}$$

wobei

$$t_{\text{Paket}} = \frac{n}{B}$$
$$t_{\text{Ack}} = \frac{n_{\text{Ack}}}{B}$$

(b) Für  $n \rightarrow \infty$  geht  $\eta \rightarrow 1$ , daher sollten Pakete möglichst groß sein.

Antworten (a) und (b) berücksichtigen Paketfehler nicht. Die Paketfehlerwahrscheinlichkeit steigt aber mit wachsender Paketlänge (siehe Aufgabenblatt 4) – d.h. eine größere Paketlänge kann sich auch negativ auf die Kanaleffizienz auswirken. Die optimale Paketlänge für Send-and-Wait (SnW) kann in Abhängigkeit von der Bitfehlerwahrscheinlichkeit  $p$  rechnerisch ermittelt werden (siehe folgende Aufgabe).

#### Musterlösung zu Aufgabe 3:

Sei  $p$  die Bitfehlerwahrscheinlichkeit. Dann gilt für die Paketfehlerwahrscheinlichkeit  $\varepsilon$

$$\varepsilon = 1 - (1 - p)^n$$

und nach Aufgabe 2c, Blatt 4 ist dann  $\frac{1}{1-\varepsilon}$  die mittlere Anzahl an Paketübertragungen. Die Kanaleffizienz  $\eta$  im fehlerbehafteten Kanal ist damit um diesen Faktor geringer als im fehlerfreien Fall:

$$\eta = (1 - p)^n \frac{t_{\text{Paket}}}{t_{\text{Paket}} + \tau + t_{\text{Ack}} + \tau}$$

Aus dieser Formel wird ersichtlich, dass im Gegensatz zur vorhergehenden Aufgabe große Pakete nicht nur Vorteile bringen, siehe auch die Bilder 1 und 2

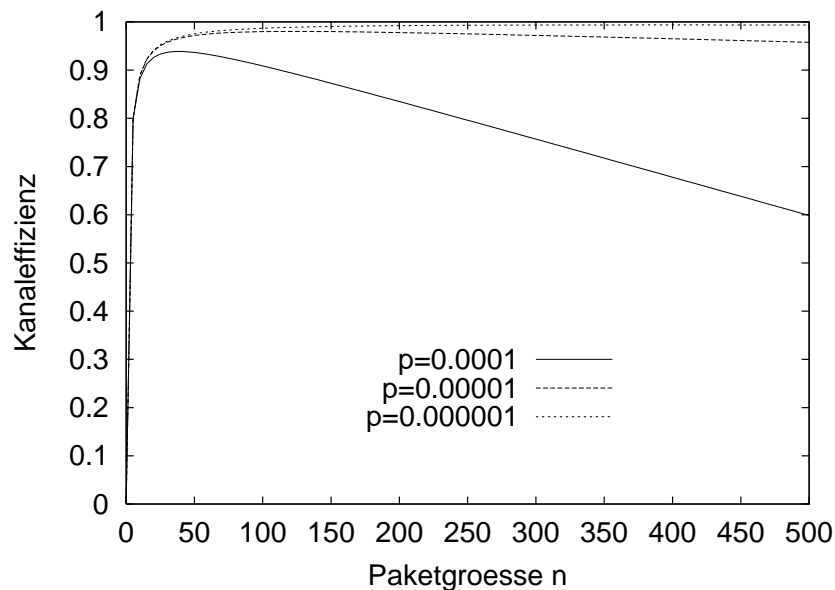


Abbildung 1: Kanaleffizienz für variierendes  $n$  bei unterschiedlichen Fehlerraten  $p$ ,  $B\tau = 1$

Es existiert also ein klares Optimum.

#### Musterlösung zu Aufgabe 4:

Anstatt eines großen Paketes schickt man viele kleine und wiederholt nur bei Bedarf (Go-back-N (GbN), Selective Repeat (SR)), idealerweise nur die verlorenen Pakete (SR).

**Sequenznummer** Pakete werden durchnummeriert, wobei nur begrenzte Anzahl an Sequenznummern zur Verfügung steht  $\rightarrow$  Sequenznummernraum

**Übertragungsfenster** Die Größe des Übertragungsfensters gibt auf der Senderseite an, wie viele Pakete (bzw. welche Sequenznummern) ohne Empfangsbestätigung versendet werden können. Auf der Empfängerseite werden die als nächstes zu empfangenen Pakete (bzw. deren Sequenznummern) festgelegt.

**Timer** Ein Timer legt die Zeitdauer fest, in der eine Empfangsbestätigung vorliegen muss, bevor ein Paket erneut gesendet wird.

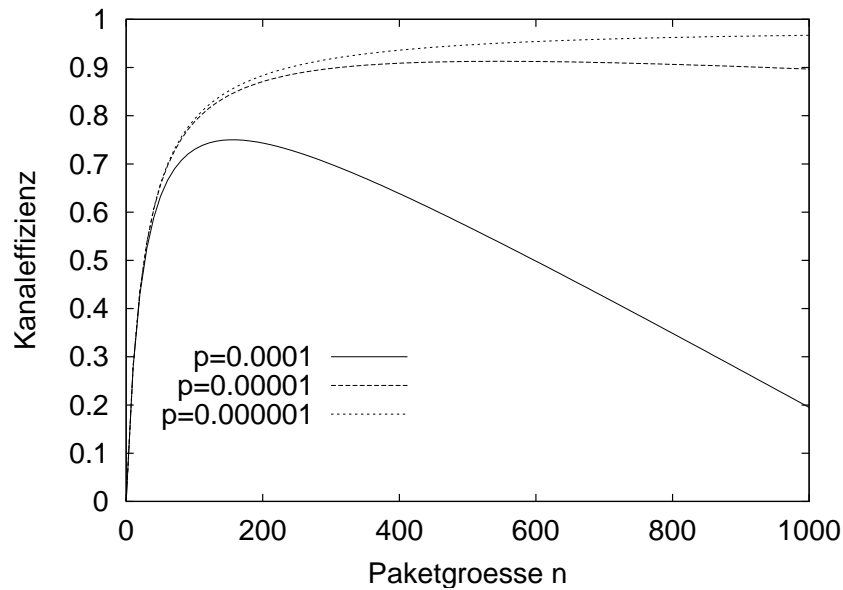


Abbildung 2: Kanaleffizienz für variierendes  $n$  bei unterschiedlichen Fehlerraten  $p$ ,  $B\tau = 100$

### Musterlösung zu Aufgabe 5:

Da die Darstellung von Sequenznummern Platz im Paketkopf benötigt, ist es aus Effizienzgründen wichtig, nur die minimale Anzahl von Sequenznummern zu benutzen. Grundsätzlich gilt, daß die benötigte Anzahl an Sequenznummern sowohl vom verwendeten ARQ-Verfahren als auch von der möglichen Fenstergröße beim Empfänger und Sender abhängt. Insbesondere ist wesentlich, dass der Empfänger in der Lage sein muss, alte von neuen Paketen zu unterscheiden und der Sender in der Lage ist, Bestätigungen (ACKs) zu unterscheiden.

(a) Für das Übertragungsfenster  $w$  muss gelten:

- Für SR:  $2 \cdot w \leq s$
- Für GbN:  $w < s$

(b) **Go-Back-N:**

Für eine Fenstergröße  $w$  wird ein Sequenznummernraum  $s \geq w + 1$  benötigt; d.h. für  $w = 4$  ist  $s = 4$  zu klein,  $s = 5$  aber ausreichend.

- Gegenbeispiel für  $s = 4$  bei  $w = 4$ : Siehe Abbildung 3.
- Begründung für  $s > w$ : Ist der Sequenznummernraum echt größer als die Fenstergröße kann der Empfänger stets unterscheiden, ob es sich um eine Wiederholung eines Paketes oder um ein neues Paket handelt – es stehen genügend Nummern zur Verfügung, um diese Unterscheidung zu erlauben.

- **Selective Repeat:**

Für eine Fenstergröße  $w$  wird ein Sequenznummernraum  $s \geq 2w$  benötigt; d.h. für  $w = 4$  ist  $s = 7$  zu klein,  $s = 8$  ausreichend.

- Gegenbeispiel für  $s = 7$ : Siehe Abbildung 4. Zwei unterschiedliche Fehlerszenarien führen beim Empfänger zur gleichen beobachteten Ab-



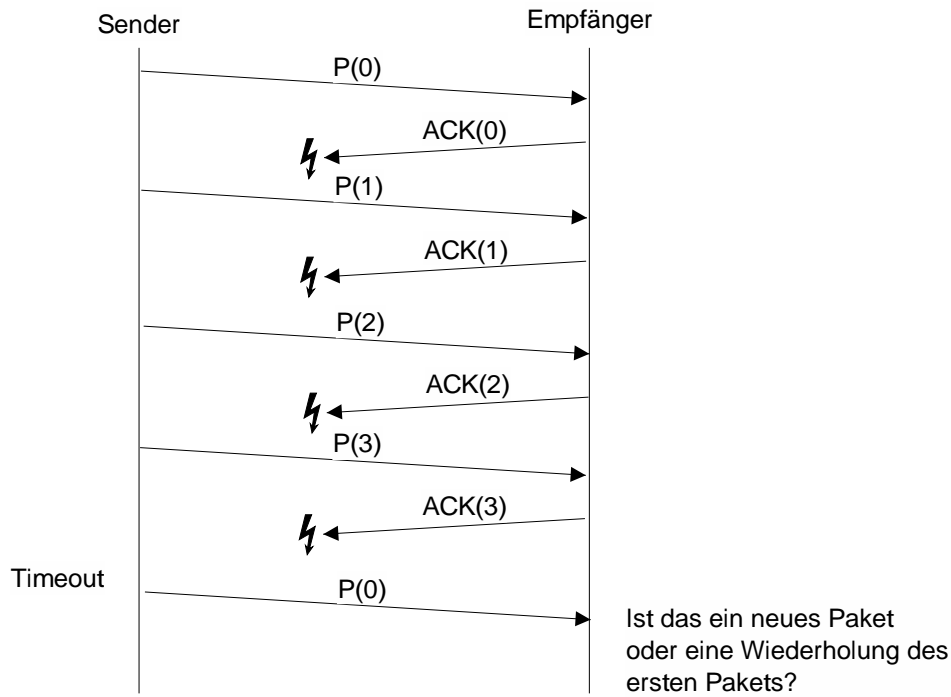


Abbildung 3: Zu kleiner Sequenznummernraum für Go-Back-N

folge von Sequenznummern. Damit kann der Empfänger nicht zwischen diesen unterschiedlichen Fehlerfällen unterscheiden, d.h. das Verfahren ist unzureichend.

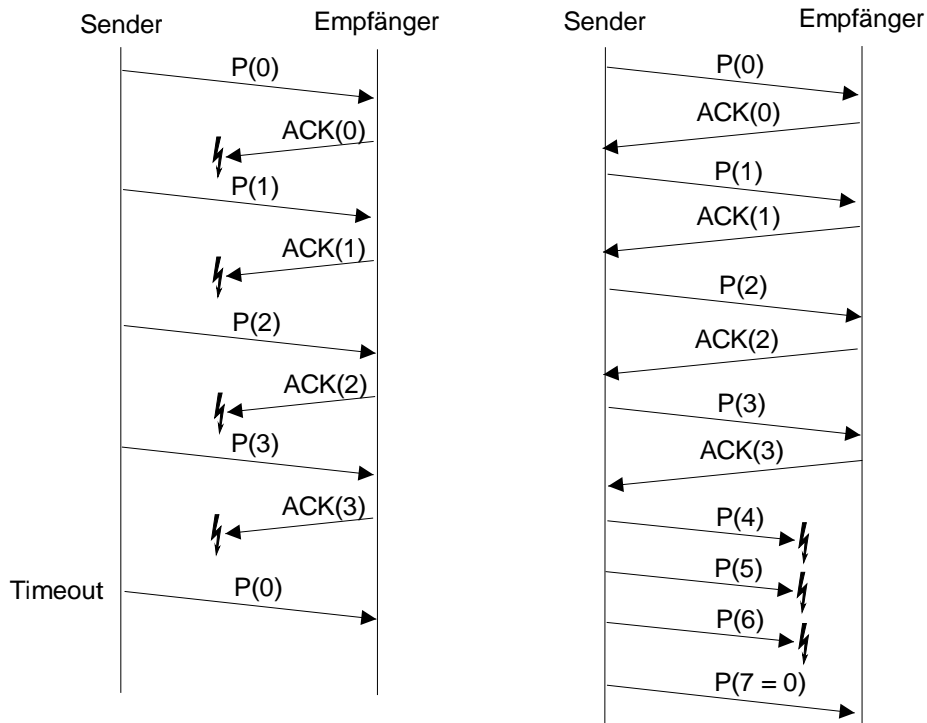


Abbildung 4: Zu kleiner Sequenznummernraum für Selective Repeat

- Begründung für  $s \geq 2w$ : Werden  $w$  Pakete gesendet und gehen möglicherweise alle Bestätigungen verloren, so muss der Empfänger in der La-

ge sein zu entscheiden, welche der folgenden Pakete Wiederholungen und welche neue Pakete darstellen. Dies ist nur möglich, wenn für das nächste Fenster ein neuer Satz an Sequenznummern zur Verfügung steht.

**Musterlösung zu Aufgabe 6:**

Nein. Das Problem ist der Verbindungsaufbau: wie kann ein Sender erkennen, dass der Empfänger das erste Paket erhalten hat? Schweigen des Empfängers kann zwei Sachen heißen: Paket erhalten bzw. Leitung tot.

## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 6

#### — Medienzugriff —

#### **Musterlösung zu Aufgabe 1:**

Mögliche Klassifikationsmerkmale:

1. Kollisionen (a) können auftreten, (b) sind ausgeschlossen
2. (Zeit-)Synchronisation der Stationen (a) notwendig, (b) nicht notwendig
3. Rückhörmöglichkeit während des Sendens (a) notwendig, (b) nicht notwendig
4. Eine Mindestbandbreite pro Station ist (a) garantiert, (b) nicht garantiert

Zuordnung der einzelnen MAC-Protokolle:

- ALOHA: 1a, 2b, 3b, 4b
- slotted ALOHA: 1a, 2a, 3b, 4b
- non-persistent CSMA: 1a, 2b, 3b, 4b
- p-persistent CSMA: 1a, 2a, 3b, 4b
- CSMA/CD: 1a, 2b, 3a, 4b
- TDMA: 1b, 2a, 3b, 4a
- Token Ring: 1b, 2b, 3b, 4a
- MACA: 1a, 2b, 3b, 4b

#### **Musterlösung zu Aufgabe 2:**

- ALOHA: Station sendet, sobald Daten vorliegen, bei einer Kollision (z.B. erkannt durch fehlendes ACKs) wartet die Station ein zufällige Zeitdauer und wiederholt die Übertragung.
- slotted ALOHA erweitert "pure ALOHA: Stationen sind synchronisiert, und dürfen nur zu Beginn eines Zeitabschnitts (slot) senden. Pakete haben maximal die Länge eines slots.
- Der Konkurrenzzeitraum (vulnerable period) wird mit slotted ALOHA halbiert, d.h. die Mglk. einer Kollision entsprechend reduziert.

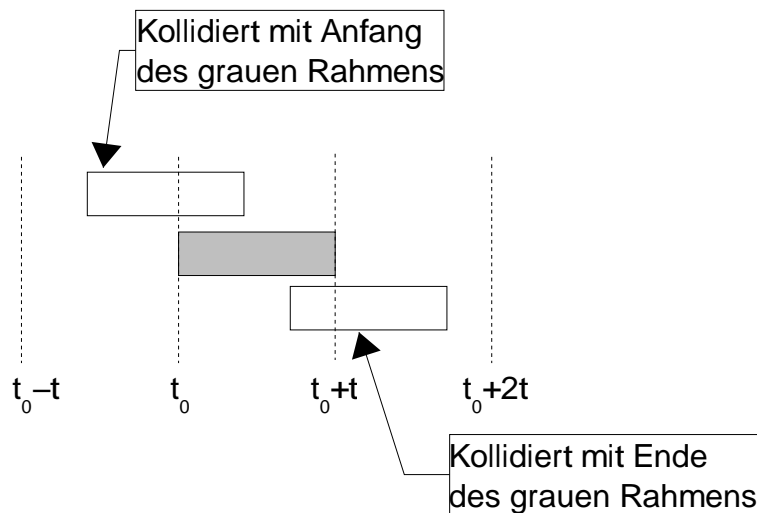


Abbildung 1: Gefährdungszeitraum eines ALOHA-Rahmens

**Musterlösung zu Aufgabe 3:**

- $G$  ist die angebotene Last, die von allen  $n$  Sendern ( $n \rightarrow \infty$ ) erzeugt wird, gemessen in Paketen pro Paketdauer.
- Der Konkurrenzzeitraum ist die Zeit, in der sich zwei Pakete bei der Übertragung stören können. (siehe 2).
- Damit das Paket, das gerade von einer Station gesendet wird, erfolgreich ist, müssen alle anderen Stationen während des gesamten Konkurrenzzeitraumes (mit einer angebotenen Last von  $2G$ ) schweigen, d.h. es wird kein weiteres Paket verschickt:

$$P_2(K = k) = \frac{(2G)^k e^{-2G}}{k!}$$

für  $k = 0$  folgt:

$$P_2(K = 0) = e^{-2G}$$

- Durchsatz  $S$  ist die angebotene Last  $G$  multipliziert mit der Wahrscheinlichkeit für das Gelingen der Übertragung  $P_2(K = 0)$ . In einem Paketzeitraum werden  $G$  Pakete verschickt und mit der Wahrscheinlichkeit  $P_2(K = 0)$  stört keine andere Station die Übertragung, d.h.:

$$\begin{aligned} S &= GP_2(K = 0) \\ &= Ge^{-2G} \end{aligned}$$

⇒ maximaler Durchsatz von ALOHA:  $S = 0.18$  (bei  $G = 0.5$ )

Analog für slotted ALOHA:  $S = Ge^{-G}$  ⇒ maximaler Durchsatz von slotted ALOHA:  $S = 0.37$  (bei  $G = 1$ )

**Musterlösung zu Aufgabe 4:**

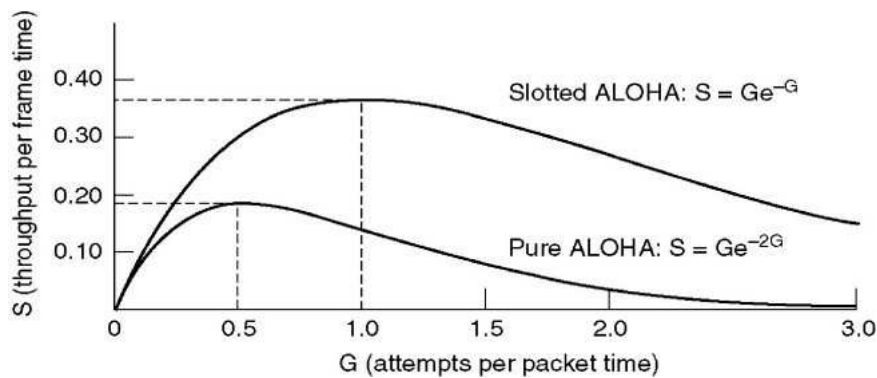


Abbildung 2: Durchsatz ALOHA/slotted ALOHA

- **persistent CSMA:** Jede Station, die ein Paket senden möchte, beobachtet den Übertragungskanal. Falls dieser frei ist, dann sendet die Station ihr Paket. Kommt es während der Übertragung des Pakets zu einer Kollision (erkennbar an der fehlenden Quittierung des Pakets), dann wartet jede Station, deren Paket mit den Paketen anderer Stationen kollidiert ist, eine zufällige Zeit bis zum erneuten Übertragungsversuch. Falls der Übertragungskanal belegt ist, dann wartet die Station so lange, bis der Übertragungskanal wieder frei ist, und sendet dann ihr Paket.

Warten zwei Stationen auf das Ende der Übertragung einer dritten Station, kommt es am Ende dieser Übertragung mit Sicherheit zur Kollision.

- **nonpersistent CSMA:** Diese CSMA-Variante arbeitet bis auf eine Ausnahme wie persistent CSMA: Falls der Übertragungskanal belegt ist, dann wartet die Station eine zufällige Zeit und beobachtet erst dann wieder den Übertragungskanal. Bei nonpersistent CSMA sind die Stationen bei einem belegten Übertragungskanal etwas weniger „gierig“.
- **$p$ -persistent CSMA:** Der Übertragungskanal ist bei  $p$ -persistent CSMA in einzelne Zeitslitze (Timeslots) aufgeteilt, die nicht notwendigerweise der Dauer einer Paketübertragung maximaler Paketlänge entsprechen müssen. Jede Station, die ein Paket senden möchte, beobachtet den Übertragungskanal. Falls der aktuelle Slot frei ist, dann sendet die Station mit der Wahrscheinlichkeit  $p$  ihr Paket oder wartet mit der Wahrscheinlichkeit  $q = 1 - p$  auf den nächsten Slot. Kommt es während der Übertragung des Pakets zu einer Kollision mit Paketen anderer Stationen, dann wartet die Station, deren Paket kollidiert ist, eine zufällige Zeit und wiederholt den Übertragungsversuch. Falls der Kanal belegt ist, dann wartet die Station auf den nächsten freien Slot, der wiederum mit der Wahrscheinlichkeit  $p$  für die Übertragung des Pakets genutzt wird.

Zusatzfragen: Gibt es ein  $p$ , so dass  $p$ -persistent CSMA gleich persistent CSMA ist? Welche Konsequenzen hat  $p$  hinsichtlich des Verhaltens bei hoher Last, welche Auswirkungen auf die mittlere Verzögerungszeit?

#### Musterlösung zu Aufgabe 5:

Medienzugriff bei Ethernet (IEEE 802.3):

- Der Standard IEEE 802.3 spezifiziert (persistent-)CSMA/CD als Protokoll für den Medienzugriff. Hierbei wird eine Kollision schon während des Sendens erkannt und die Übertragungen mittels eines Störungssignals (*jamming*)

signal) abgebrochen. Die Wartezeit nach einer Kollision wird zufällig nach dem Mechanismus *Exponential Binary Backoff* bestimmt:

- Jede Station wählt nach der  $i$ -ten Kollision,  $0 \leq i \leq 16$ , zufällig eine Zahl  $k$  aus dem Intervall  $[0, 2^i - 1]$ ,  $i \leq 10$ , bzw.  $[0, 1023]$ ,  $10 < i \leq 16$  und wartet  $k$  Zeitslots bis nächsten Zugriff auf das Medium.
- Größe der Zeitslots entspricht der maximalen, doppelten Übertragungsverzögerung (round-trip propagation delay,  $51,2 \mu\text{s}$  nach IEEE 802.3).

Lösung:

- Sei Zufallsvariable  $X$  definiert als Anzahl von Kollisionen vor erfolgreicher Übertragung. Gesucht ist  $P(X = 3)$ .
- Für die Wahrscheinlichkeiten, dass sich zwei anfangs gleichzeitig senden wollende Stationen im  $n$ -ten Versuch behindern gilt:

$$n = 1 \Rightarrow i = 0, \quad k \in [0, 2^0 - 1] \Rightarrow P(\text{Kollision}|n = 1) = p_1 = 1$$

$$n = 2 \Rightarrow i = 1, \quad k \in [0, 2^1 - 1] \Rightarrow P(\text{Kollision}|n = 2) = p_2 = \frac{1}{2}$$

$$n = 3 \Rightarrow i = 2, \quad k \in [0, 2^2 - 1] \Rightarrow P(\text{Kollision}|n = 3) = p_3 = \frac{1}{4}$$

$$n = 4 \Rightarrow i = 3, \quad k \in [0, 2^3 - 1] \Rightarrow P(\text{Kollision}|n = 4) = p_4 = \frac{1}{8}$$

⋮

- Damit folgt:

$$P(X = 3) = p_1 \cdot p_2 \cdot p_3 \cdot (1 - p_4) = 0.109375 \approx 11\%.$$

### Musterlösung zu Aufgabe 6:

Gängige Radios mit nur einer Antenne können nicht gleichzeitig senden und den Kanal überwachen (empfangen). Full-duplex/Mehrfach-Antennen Radios sind extrem aufwendig und daher kaum verbreitet.

Weitere Probleme (für CSMA-Verfahren allgemein) ergeben sich bei der Überprüfung des Kanals:

- Ein freier Kanal am Sender bedeutet nicht notwendigerweise, dass der Kanal am Empfänger ebenfalls frei ist (hidden terminal)
- Ein belegter Kanal am Sender bedeutet nicht notwendigerweise, dass der Kanal am Empfänger ebenfalls belegt ist (exposed terminal)

## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 7

#### — Routing und Forwarding —

#### Musterlösung zu Aufgabe 1:

- (a) Leitungsvermittlung (circuit switching), s. Abb. 1. Die gesuchte Zeit setzt sich aus der für den Verbindungsaufbau benötigten Zeit, der Sendezeit und der Laufzeit von A nach D zusammen:

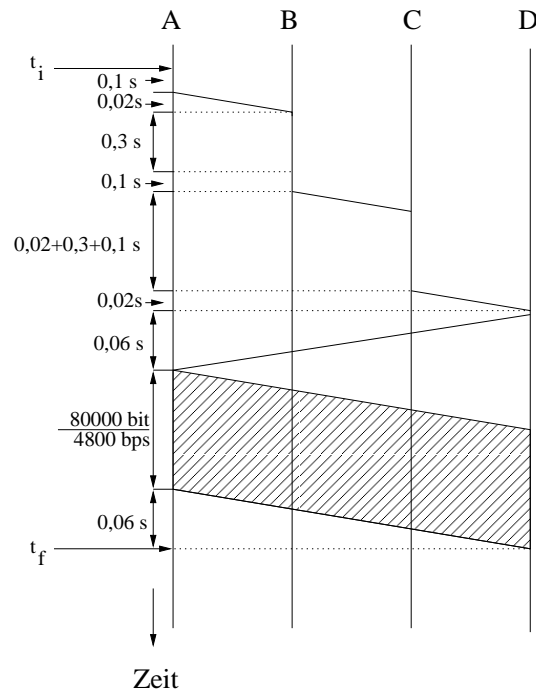


Abbildung 1: Weg-Zeit-Diagramm für circuit switching

$$t_{\text{ges}} = t_f - t_i = 2 \cdot 0,3 \text{ s} + 3 \cdot 0,1 \text{ s} + 9 \cdot 0,02 \text{ s} + \frac{80000 \text{ bit}}{4800 \text{ bps}} \approx 17,75 \text{ s}$$

- (b) Nachrichtenvermittlung (message switching), s. Abb. 2. Die gesuchte Zeit setzt sich aus der Sendezeit bei A, B und C, der Laufzeit von A→B, B→C und C→D, und der Speicherzeit bei B und C zusammen:

$$t_{\text{ges}} = t_f - t_i = 3 \cdot \frac{8 \cdot 10050 \text{ bit}}{4800 \text{ bps}} + 3 \cdot 0,02 \text{ s} + 2 \cdot 0,3 \text{ s} = 50,91 \text{ s}$$

- (c) Paketvermittlung (packet switching), s. Abb. 3. Die 10.000 Zeichen werden in 20 Pakete aufgeteilt. Jedes Paket umfaßt 500 Zeichen und einen zusätzlichen Header von 50 Zeichen. Somit ist die Gesamtlänge eines Paketes 550

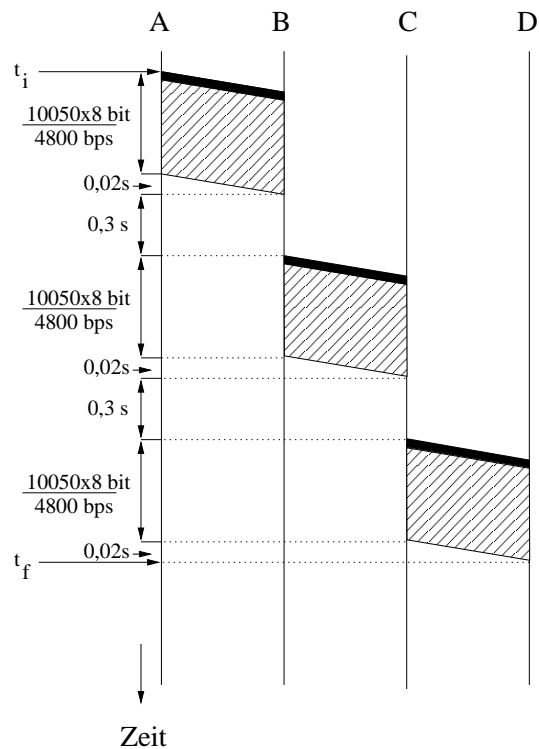


Abbildung 2: Weg-Zeit-Diagramm für message switching

Zeichen. Die Zwischenknoten B und C speichern jedes Paket 0,3 Sekunden lang und senden es anschließend weiter. Somit sendet B das erste Paket schon weiter zu C während A noch die letzten der 20 Pakete sendet. Beachte, daß das der wesentliche Unterschied zwischen message und packet switching ist (beim message switching muß ein Zwischenknoten erst die *gesamte* Nachricht erhalten haben bevor sie zum nächsten Knoten weitergesendet wird.). Beachte, daß Knoten A, B und C alle 20 Pakete unmittelbar nacheinander senden. Dieser Effekt ist bekannt als *Pipelining*.

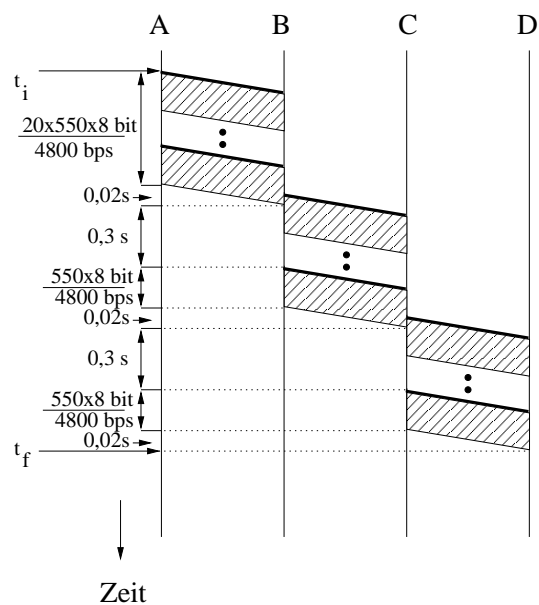


Abbildung 3: Weg-Zeit-Diagramm für packet switching



Die gesuchte Zeit setzt sich aus der Sendezeit der 20 Pakete bei A, der Speicherzeit und Sendezeit des letzten der 20 Pakete bei B und C, und der Laufzeit von A→B, B→C und C→D zusammen:

$$t_{\text{ges}} = t_f - t_i = 20 \cdot \frac{8 \cdot 550 \text{ bit}}{4800 \text{ bps}} + 3 \cdot 0,02 \text{ s} + 2 \cdot \left( 0,3 \text{ s} + \frac{8 \cdot 550 \text{ bit}}{4800 \text{ bps}} \right) \approx 20,83 \text{ s}$$

(d) Leitungsvermittlung ist vorteilhaft, wenn

- Längere Datenströme übertragen werden
- Die Datenströme eine konstante Bitrate benötigen
- Die Verzögerungen während der Kommunikation möglichst konstant sein müssen.

Paketvermittlung ist vorteilhaft, wenn

- Die Datenströme eher kurz sind
- Die Datenströme eine stark schwankende Bitrate aufweisen
- Schwankungen in der Verzögerung nicht zu sehr stören

Bei Paketvermittlung wird die Bandbreite nicht explizit reserviert. Erst durch den Versand eines Packets wird die Bandbreite in Anspruch genommen. Bei Leitungsvermittlung wird die Bandbreite verschwendet, wenn die im Voraus reservierte Kapazität nicht verwendet wird.

### Musterlösung zu Aufgabe 2:

Lee-Graphen stellen ein vereinfachtes Modell zur Bestimmung von Blockierungswahrscheinlichkeiten von Switches und Netzwerken dar. Bedingung für die im folgenden beschriebene Anwendung ist, dass die betrachteten Wege kantendisjunkt sind.

Die Belegungswahrscheinlichkeiten der einzelnen Links sei mit  $p_i$  gegeben, die Wahrscheinlichkeit, dass ein Link nicht belegt ist mit  $q_i = 1 - p_i$ . Für die Zusammenschaltung von Links wird zwischen parallelen und seriellen Links unterschieden:

- Parallel verlaufende Links

$$\begin{aligned} p_{\text{parallel}} &= p_1 \cdot p_2 \cdot \dots \cdot p_n \\ &= \prod_{i=1}^n p_i \end{aligned}$$

- Seriell verlaufende Links

$$\begin{aligned} p_{\text{seriell}} &= 1 - q_1 \cdot q_2 \cdot \dots \cdot q_n \\ &= 1 - \prod_{i=1}^n (1 - p_i) \end{aligned}$$

Für die Lösung der Aufgaben werden zunächst die seriellen Verbindungen berechnet.

(a) Serielle Verbindungen:

$$\begin{aligned} p_1 \text{ und } p_4 &= 1 - (1 - 0,1)(1 - 0,5) = 0,55 \\ p_2 \text{ und } p_5 &= 1 - (1 - 0,2)(1 - 0,8) = 0,84 \end{aligned}$$

$$p_3 \text{ und } p_6 = 1 - (1 - 0,1)(1 - 0,5) = 0,55$$

Blockierungswahrscheinlichkeit  $B$  ergibt sich aus den drei Parallel-Links  
 $\Rightarrow B = 0,55 \cdot 0,84 \cdot 0,55 \approx 0,25$

(b) Serielle Verbindung:

$$p_1 \text{ und } p_2 = 1 - (1 - 0,1)(1 - 0,2) = 0,28$$

Blockierungswahrscheinlichkeit  $B$  ergibt sich aus den zwei Parallel-Links  
 $\Rightarrow B = 0,28 \cdot 0,1 = 0,028$

### Musterlösung zu Aufgabe 3:

- In einem LS-Verfahren sammelt jeder Router Informationen über die Kosten, um seine Nachbarrouter zu erreichen, und teilt diese Informationen allen anderen Routern des Netzes mit. Bei einem LS-Verfahren werden wenige Informationen über große Entfernungen übertragen. Jeder Router ist aber in der Lage, die Netzwerktopologie nachzubilden und mit diesem Wissen selbständig die optimalen Wege zu allen anderen Routern zu bestimmen.
- In einem DV-Verfahren sammelt jeder Router Informationen über die Kosten, um alle anderen Router des Netzes zu erreichen in einer Routingtabelle, und teilt diese Informationen seinen Nachbarroutern mit. Bei einem DV-Verfahren werden viele Informationen über kurze Entfernungen übertragen. Kein Router hat das komplette Wissen über die Netzwerktopologie, die Routinginformationen werden iterativ aktualisiert.

Gemeinsam haben beide Ansätze, dass auch in großen Netzen mit vielen Routern die kostengünstigsten Wege bestimmt werden können. Bei sich dynamisch ändernden Kosten der Links in einem Netz können die Verfahren periodisch durchgeführt bzw. die Information aktualisiert werden, um die Routing-Tabellen der Router an die neuen Gegebenheiten im Netz anzupassen.

LS-Verfahren zeichnen sich durch bessere Skalierbarkeit und ein besseres Konvergenzverhalten (Count-to-Infinity Problem bei DV-Verfahren) aus.

Ein Vertreter der LS-basierten Verfahren ist beispielsweise OSPF (*Open Shortest Path First*), das auf dem Dijkstra-Algorithmus basiert. Vertreter der DV-basierten Verfahren sind RIP (*Routing Information Protocol*) und IGRP (*Interior Gateway Routing Protocol*), die auf dem Bellman-Ford-More Algorithmus basieren.

- Nach Beendigung eines der beiden Algorithmen sind für den Router, in dem der Algorithmus abgelaufen ist, die kostengünstigsten Wege zu allen anderen Routern bekannt. Wird nun der Algorithmus simultan in allen Routern des Netzes durchgeführt, dann kennen nach Beendigung des Algorithmus' alle Router jeweils die kostengünstigsten Wege zu den anderen Routern. Diese so gewonnenen Informationen werden dann in den Routing-Tabellen der Router vermerkt.
- Bei sich dynamisch ändernden Kosten der Links in einem Netz müssen die Routing-Algorithmen periodisch durchgeführt werden, um die Routing-Tabellen der Router an die neuen Gegebenheiten im Netz anzupassen.

### Musterlösung zu Aufgabe 4:

Siehe Aufgabenblatt 8, Aufgabe 1.

## Übung zur Vorlesung Kommunikationsnetze

### Aufgabenblatt 8

#### — Routing / Überlastabwehr —

#### Musterlösung zu Aufgabe 1:

Beide Algorithmen werden verwendet, um von einem Netzwerkknoten die geeignetste Verbindung zu allen anderen Knoten zu bestimmen. Dafür werden den Links zwischen den Knoten Gewichte zugeordnet. Je geringer die Summe aller Gewichte, desto besser der Pfad. Für den Dijkstra-Algorithmus sind nur positive Gewichte zulässig, Bellman-Ford-Moore erlaubt auch negative Gewichte aber keine negativen Zyklen.

(a) Beschreibung des Dijkstra Algorithmus:

N: Menge aller Knoten im Netzwerk

M: Menge der bearbeiteten Knoten

s: Knoten, dessen Routingtabelle erstellt wird

$d_{ij}$ : Link-Kosten zwischen Knoten i und j

$D_n$ : Geringste Pfadkosten zwischen Knoten s und n

1. Initialisierung

$$M = \{s\}$$

$$D_n = d_{sn} \text{ für alle Nachbarn von } s, \text{ sonst } \infty$$

2. Schritt

Suche den benachbarten Knoten w von M, der nicht in M ist und für den  $D_w$  am geringsten ist. Füge w in M ein.

3. Schritt

Führe Update aller  $D_n$  durch, wobei gilt  $D_n = \min[D_n; D_w + d_{wn}]$ . Falls  $M \neq N$  gehe zu Schritt 2, ansonsten: Ende.

Mit dem Algorithmus erhält man das folgende Ergebnis:

	M	B	C	D	E	F	G
0	{A}	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
1	{A}	1, AB	3, AC	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
2	{A,B}	1, AB	3, AC	2, ABD	$\infty, -$	$\infty, -$	$\infty, -$
3	{A,B,D}	1, AB	3, AC	2, ABD	$\infty, -$	3, ABDF	4, ABDG
4	{A,B,C,D}	1, AB	3, AC	2, ABD	6, ACE	3, ABDF	4, ABDG
5	{A,B,C,D,F}	1, AB	3, AC	2, ABD	6, ACE	3, ABDF	(4, ABDFG)
6	{A,B,C,D,F,G}	1, AB	3, AC	2, ABD	5, ABDGE	3, ABDF	4, ABDG
7	{A,B,C,D,E,F,G}	1, AB	3, AC	2, ABD	5, ABDGE	3, ABDF	4, ABDG

(b) Beschreibung des Bellman-Ford Algorithmus:

N: Menge aller Knoten im Netzwerk

s: Knoten, dessen Routingtabelle erstellt wird

$d_{ij}$ : Link-Kosten zwischen Knoten i und j

h: Maximale Anzahl der Links

$D_n^{(h)}$ : Geringste Pfadkosten zwischen Knoten s und n unter der Bedingung, dass der Pfad nicht länger als h Links ist

1. Initialisierung

$$h = 0$$

$$D_n^{(h)} = \infty \text{ für alle } n \neq s, \text{ sonst } 0$$

2. Schritt

$$h = h + 1$$

Finde den kürzesten Weg zu allen möglichen Knoten unter der Bedingung, dass der Pfad nur h Links lang ist

$$(D_n^{(h)} = \min[D_n^{(h-1)}; D_j^{(h-1)} + d_{jn}] \text{ von allen } j, n \in N)$$

Gehe zu Schritt 2 (max.  $|N|$  Iterationen).

Der Algorithmus liefert ebenfalls das korrekte Ergebnis

h	B	C	D	E	F	G
0	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
1	1, AB	3, AC	$\infty, -$	$\infty, -$	$\infty, -$	$\infty, -$
2	1, AB	3, AC	2, ABD	6, ACE	$\infty, -$	$\infty, -$
3	1, AB	3, AC	2, ABD	6, ACE	3, ABDF	4, ABDG
4	1, AB	3, AC	2, ABD	5, ABDGE	3, ABDF	4, ABDG

Für die optimale Auslastung eines Kommunikationsnetzes kann es erforderlich sein, die Senderate der einzelnen Stationen zu regulieren. Flusskontrolle (engl. *flow control*) und Staukontrolle (engl. *congestion control*) stellen in diesem Zusammenhang die wesentlichen Ansätze dar.

### Musterlösung zu Aufgabe 2:

**Flusskontrolle** : Sender passt seine Übertragungsrate der Verarbeitungsgeschwindigkeit des Empfängers an, damit dieser nicht überlastet wird.

**Staukontrolle** : Sender passt seiner Übertragungsrate an die im Netzwerk vorhandene Kapazität an, damit die erzeugte Netzwerklast nicht zu einer Überlastung des Netzwerkes führt.

Sind Flusskontrolle und Staukontrolle in einem Protokoll realisiert, dann muß der Sender seine momentane Senderate auf das Minimum der Fluss- und Staukontrolle setzen (z.B. TCP). Die wichtigsten Kriterien, nach denen Verfahren zur Flusskontrolle bzw. Staukontrolle kategorisiert werden können, sind:

- Explizit  $\leftrightarrow$  Implizit (Feedback)
  - Implizit: Sender leitet Auftreten von Überlast aus lokalen Beobachtungen ab (Verzögerung/Ausbleiben von ACKs, Anwachsen der Sendequelle)
  - Explizit: Sender wird bei Überlast benachrichtigt (Choke Packets, Warning Bit, ACK statt Permit)
- Raten-basiert  $\leftrightarrow$  Fenster-basiert (Credit-/Window-/Feedback-based)
  - Raten-basiert: Maximale Senderate wird festgelegt

- Fenster-basiert: Sender erhält Erlaubnis für Übertragung einer bestimmten Datenmenge (Pakete) in Form von Übertragungsfenstern (trivialerweise Fenster für 1 Paket)
- Hop-by-Hop ↔ End-to-End
  - Hop-by-Hop: Alle Stationen auf dem Weg eines Paketes werden bezüglich Überlast informiert, Rückstau (Backpressure) bis zum Sender
  - End-to-End: Nur der Sender wird bezüglich Überlast informiert

### Musterlösung zu Aufgabe 3:

Aus dem Aufgabentext (ggf. mit Skizze zu ergänzen):

$\rho$ : Rate mit der Marken erzeugt werden  
 $d$ : Datemenge die eine Marke repräsentiert  
 $\sigma$ : Kapazität des Eimers

- (a) Die maximale Datenmenge  $D$  ergibt sich bei vollem Eimer aus der Kapazität  $\sigma$  und der in der Zeit  $t$  erzeugten Marken:

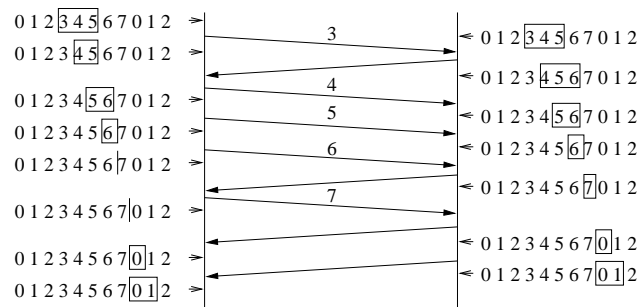
$$D = \sigma + \rho t d$$

- (b) Bei vollem Eimer kann sofort gesendet werden und daher gilt  $t_{min} = 0s$ . Bei leerem Eimer werden vier Marken benötigt und damit  $t_{max} = 4s$ .

### Musterlösung zu Aufgabe 4:

- (a) – Acknowledgements sind Quittungen für vom Empfänger korrekt empfangene Datenpakete. Permits gestatten dem Sender das Senden weiterer Datenpakete. In einigen Protokollen (z.B. TCP) werden Acknowledgements und Permits in einem Packet gemeinsam versendet.
- Das Ausbleiben von Permits sorgt für eine Verkleinerung des Fensters auf Senderseite (es müssen Mechanismen existieren, damit keine deadlocks entstehen!).

- (b) Ablauf der Fenster-basierten Übertragung:



### Musterlösung zu Aufgabe 5:

TCP-Überlastabwehr beinhaltet Fluss- und Staukontrolle:

- Flusskontrolle: RCV WND (Receive Window) mit Empfänger vereinbart
- Staukontrolle: CWND (Congestion Window) initial 1 MTU (Max. Transmission Unit)

⇒  $AWND (Allowed Window) = \min[RCV WND, CWND]$

Es werden zwei Phasen unterschieden: *Slow Start* und *Congestion Avoidance*. In beiden Phasen wird das CWND nach jeder erfolgreichen Übertragung vergrößert bis entweder ein Timeout erfolgt oder die Größe des RCV WND erreicht wird. Der Übergang zwischen beiden Phasen wird von der Ssthresh (Slow-Start Threshold) markiert. Tritt ein Timeout auf, wird diese auf die Hälfte des aktuellen CWND gesetzt, d.h.  $Ssthresh = CWND/2$ .

#### Slow Start:

- $CWND < Ssthresh$
- $CWND = CWND * 2$  nach jeder erfolgreichen Übertragung → exponentielles Wachstum

#### Congestion Avoidance:

- $CWND \geq Ssthresh$
- $CWND = CWND + 1 (MTU)$  nach jeder erfolgreichen Übertragung → lineares Wachstum

