Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Professor Dr. W. Effelsberg
Stephan Fischer, Raum 415

Multimediasysteme
Wintersemester 1996/97
Übungsaufgaben

Multimediasysteme: Übungsblatt 6

Aufgabe 1 *Übertragungszeiten*

Die folgende Tabelle stellt die Übertragungszeiten und die Zahl verfügbarer Puffer für Stationen eines Übertragungsweges dar. Station 5 ist der Empfänger, Station 1 der Sender. Jeder Puffer kann maximal 10 KB speichern. Ein Paket kann nur als Ganzes in einem Puffer gespeichert werden. Eine Pufferoperation dauert in jeder Station 2 ms. Bei welcher Größe n der zu übertragenden Nachricht kann eine Übertragungsdauer von 40 ms nicht mehr gewährleistet werden?

Station	Verfügbare Puffer	Leitungskapazität zur nächsten Station
1	0	200 KB/s
2	5	100 KB/s
3	1	166.66 KB/s
4	3	111.11 KB/s
5		

Lösung:

Berechnung der Paketgröße:

$\frac{xs}{200KB} + \frac{xs}{100KB} + \frac{xs}{166.66KB} + \frac{xs}{111.11KB} \leq 40ms$ ergibt

$x * \frac{100*166.66*111.11+200*166.66*111.11+200*100*111.11+200*100*166.66}{200*100*166.66*111.11} \leq 0.04KB$,

also $x \leq 1333.32Bytes$. Bei dieser Größe ist eine Pufferung nie notwendig.

b) Welche Nachrichtengröße kann pro Sekunde ohne Verwendung der Puffer übertragen werden?

Lösung:

Engpaß ist die Leitung mit 100 KB/s.

c) Wie lange dauert die Übertragung von 1000 Bytes?

Lösung:

30ms

d) Wie lange dauert die Übertragung von 200 KB? Gehen Sie davon aus, daß jede Station der vorhergehenden mittels Stop-and-wait mitteilt, daß sie ein Paket x erhalten hat. Die Dauer der Bestätigung kann vernachlässigt werden.

Lösung:

von Station nach	Paket	Paket komplett angekommen	Speicherung in in	Paket komplett übertragen
1-2	1	0.05 s	Leitung nach 3	0.15 s
1-2	2	0.1 s	Puffer 1	0.25 s
1-2	3	0.15 s	Puffer 2	0.35 s
1-2	4	0.2 s	Puffer 3	0.45 s
1-2	5	0.25 s	Puffer 4	0.55 s
1-2	6	0.3 s	Puffer 5	0.65 s
1-2	7	0.35 s	Puffer 1	0.75 s
1-2	8	0.4 s	Puffer 2	0.85 s
1-2	9	0.45 s	Puffer 3	0.95 s
1-2	10	0.5 s	Verlust	–
1-2	10	0.55 s	Puffer 4	1.05 s
1-2	11	0.6 s	Verlust	–
1-2	11	0.65 s	Puffer 5	1.15 s
1-2	12	0.7 s	Verlust	–
1-2	12	0.75 s	Puffer 1	1.25 s
1-2	13	0.8 s	Verlust	–
1-2	13	0.85 s	Puffer 2	1.35
1-2	14	0.9 s	Verlust	–
1-2	14	0.95 s	Verlust	–
1-2	14	1.0 s	Puffer 3	1.45
1-2	15	1.05 s	Verlust	–
1-2	15	1.1 s	Puffer 4	1.55
1-2	16	1.15 s	Verlust	–
1-2	16	1.2 s	Puffer 5	1.65
1-2	17	1.25 s	Verlust	–
1-2	17	1.3 s	Puffer 1	1.75 s
1-2	18	1.35 s	Verlust	–
1-2	18	1.4 s	Puffer 2	1.85 s
1-2	19	1.45 s	Verlust	–
1-2	19	1.5 s	Puffer 3	1.95
1-2	20	1.55 s	Verlust	–
1-2	20	1.6 s	Puffer 4	2.05

Dabei wird die Ausgangsleitung nur benutzt, wenn sie frei ist bzw. wenn keine Puffer belegt sind (FIFO). Das Einspeisen in den Puffer dauert 2 ms. Da die Dauer der Bestätigung klein ist, findet die Retransmission jeweils im nächsten Zeitintervall statt. Da nach Station 2 höchstens eine Geschwindigkeit von 100 KB/s erreicht wird, langt es, das letzte Paket aus Station 3 hochzurechnen. Dies ergibt eine Zeit von $2.05 + 0.06 + 0.09 = 2.2s$.

Aufgabe 2 *Video-on-Demand*

Betrachten Sie eine Video-on-Demand Datenbank. Diese bestehe aus einem Verbund von Filmservern und geeigneten Speichermedien. Wie sollten Filme in dieser Datenbank gespeichert werden, um Applikationen möglichst schnell zu bedienen? Überlegen Sie, welche Hilfsmittel im burst-case angewendet werden können.

Lösung:

Um eine hohe Speicherungseffizienz und Bandbreite zu erreichen, kombiniert ein Videoserver normalerweise mehrere Platten in einem Diskarray. Jede dieser Disks hat genug Bandbreite, um mehrere Benutzer, die ein Video abrufen, zu bedienen. Um die Benutzung des Disksystems zu optimieren, müssen mehrere Dinge bedacht werden:

- die Retrievalmethode: Die Retrievalmethode bestimmt, wie ein Datenstrom von der Disk gelesen wird. Sie ist verantwortlich, den Clients Deadlines zu garantieren und diese einzuhalten
- die Speicherbelegung: Um intensive Suchbewegungen der Diskköpfe zu vermeiden, müssen die Blöcke auf der Disk entsprechend dem Videostream angeordnet werden. Hier muß ein Kompromiß zwischen Suchbewegung und der Bedienung von mehreren Benutzern gefunden werden.
- das Striping: Die gespeicherten Daten müssen über mehrere Disks verteilt werden. Dies verhindert sog. *hotspots*, wo einzelne Disks stark beansprucht werden und andere wenig zu tun haben. Daher sind die Daten eines Stroms normalerweise über alle Disks verteilt
- Videoserver-Arrays: die Benutzung eines solchen Arrays garantiert, daß jeder Server nur einen Teilstrom bereitstellt. Dies verhindert sog. hot-spot-Server.

Burst case: In der Datenbank sollte die empirical envelope $\epsilon(\tau)$ vorberechnet werden. Mit Hilfe dieser Funktion können die Puffer optimal geladen werden und so ein burst case optimal behandelt werden.

Aufgabe 3 *Empirical Envelope*

Geben Sie einen Algorithmus in Pseudocode an, der unter Ausnutzung einer vorberechneten *empirical Envelope* die vorhandenen Systempuffer optimal bedient.

Lösung:

Empirical Envelope: $\epsilon(\tau) = max_t A_i[t, t+\tau], \forall t \in [0, T_{Total} - \tau]$

Wenn τ die Wiedergabezeit eines Frames ist, ist $\frac{\epsilon(\tau)}{\tau}$ die Peak-Bitrate des Stroms. Ist τ die Wiedergabezeit des gesamten Videos, dann ist $\frac{\epsilon(\tau)}{\tau}$ die mittlere Bitrate des Stroms.

Abruf der Videos in zwei Techniken:

- Constant time length(CTL)
- Constant data length(CDL)

.

Algorithmus

```
if (Wiedergabe == CTL){
    Belege Systempuffer, die ε(τ) Bytes speichern können
    Für jedes Zeitintervall
        Lese gewünschte Daten
}
else { /* Wiedergabe ist CDL */
    Allokiere 2 ∗ ε(τ) Pufferplatz
    Für jedes Zeitintervall
        if (Daten in Puffer ≥ ε(τ))
            Wait until next round
        else {
            Read ε(τ) data block
        }
}
```