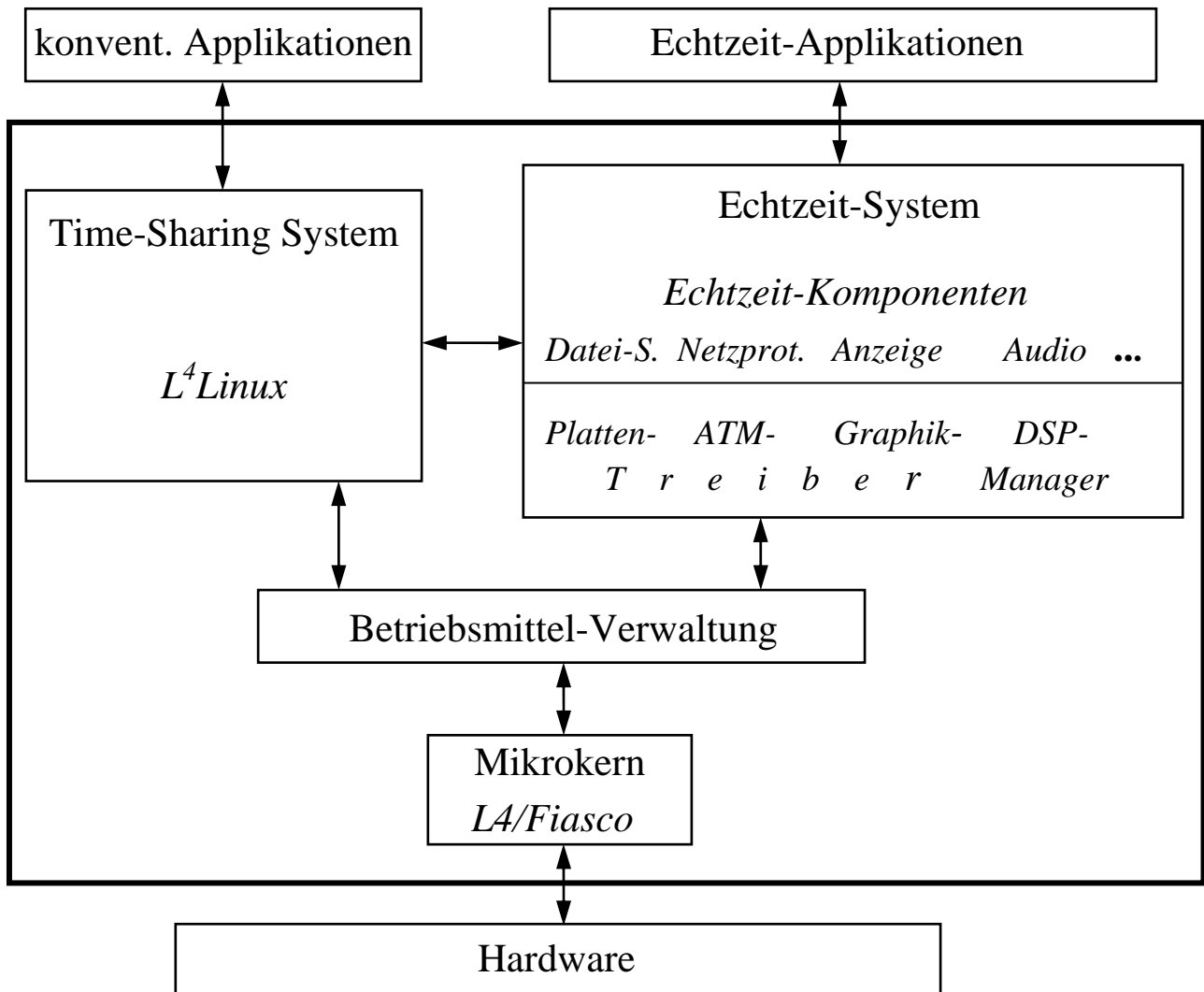


# 7. Quality-Assuring Scheduling

## 7.1. Ausgangspunkt

- Motivation

– *Architektur von DROPS – Dresden Real-Time OPerating System*



<http://os.inf.tu-dresden.de/drops/overview.html>

– *Arbeitslast*

periodische, unabhängige Tasks

(stark) schwankender Ressourcenbedarf

„Wichtigkeit“ innerhalb (und zwischen) Tasks

- **Hintergrund**

- **Überbuchung**

Imprecise Computations

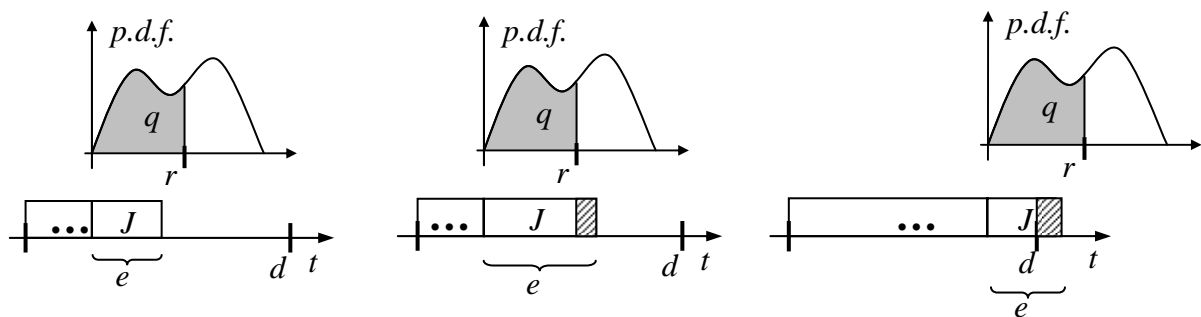
SRMS

- **DROPS Scheduler**

Scheduling mit festen Prioritäten

Reservierungsprioritäten (4 Ebenen)

- **Idee**



## 7.2. Task-Modell

- **Task**

Folge von Jobs, bestehend aus Pflicht- und Wahlteil  $M_i, O_i$

- **Taskbeschreibung**

$$\tau_i = (X_i, Y_i, w_i, q_i, t_i)$$

$X_i$ : Zufallsvariable; Ausführungszeit Pflichtteil

$Y_i$ : Zufallsvariable; Ausführungszeit Wahlteil

$w_i$ : maximale Ausführungszeit Pflichtteil

$q_i$ : QoS-Parameter

$t_i$ : Periode = Deadline

### 7.3. Scheduling und Admission für Tasks mit einheitlicher Periodenlänge

- **Allgemeines Vorgehen**

- **Zuordnungen**

Task  $\tau_i \mapsto pr(M_i), pr(O_i)$  feste Prioritäten

Task  $\tau_i \mapsto r_i$  Reservierungszeit für Wahlteil

$$p_i(r) := \mathbf{P}(Y_i \leq r \wedge O_i \text{ ist spätestens bei } t_i \text{ beendet}), \quad r \in \mathbb{R}$$

Reservierungszeit  $r_i$  für  $\tau_i$ :

$$r_i = \min(r \in \mathbb{R} \mid p_i(r) \geq q_i) \quad \forall i = 1, \dots, n \quad (*)$$

- **Zulassung für  $T = \{\tau_1, \dots, \tau_n\}$**

- (1) Alle  $X_i$  müssen ihre Deadline einhalten.
- (2) Das Gleichungssystem (\*) besitzt eine Lösung.

- **Einheitliche Periodenlänge  $t$**

- **Zulassung nach (1)**

$$\sum_{i=1}^n \frac{w_i}{t} \leq 1$$

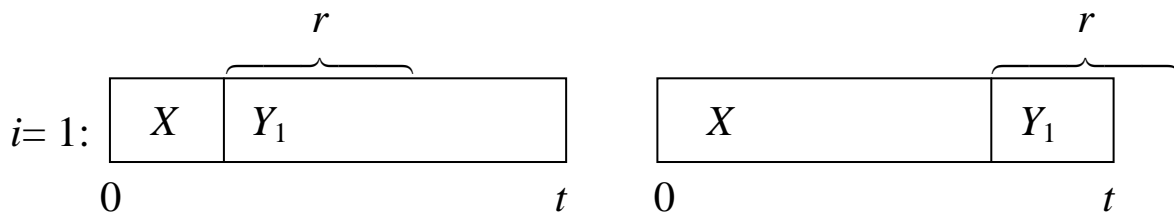
- **Prioritätszuordnung**

„qualitätsmonoton“ (QMS):  $q_1 \geq q_2 \geq \dots$

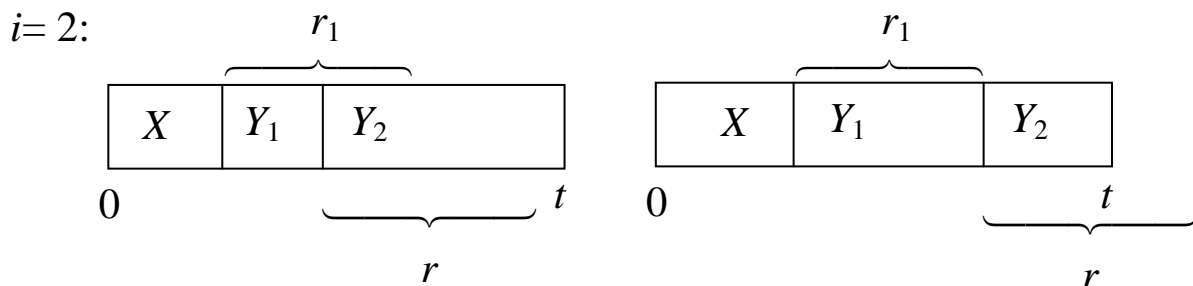
QMS ist optimal?

- **Reservierungszeit**

$$X := \sum_{i=1}^n X_i$$



$$p_1(r) =$$



$$p_2(r) =$$

$$p_i(r) = P\left(Y_i \leq r \wedge X + Y_i + \sum_{k=1}^{i-1} \min(Y_k, r_k) \leq t\right)$$

– **Beispiel**

$n = 2$  Anzahl Tasks;

$t = 7$  Periodenlänge

$X_1, \dots, Y_2$  gleichverteilt, Werte: 1, 2, 3;

$q_1 = 90\%$

$i = 1:$	$X$		<i>Wert</i>
	$Y_1$		<i>Wkt.</i>
1	$\frac{1}{3}$		
2	$\frac{1}{3}$		
3	$\frac{1}{3}$		
<i>Wert</i>			
<i>Wkt.</i>			

– **Probleme**

Faltung – Minimum – Unabhängigkeit

– **Modifikationen**

Subjobs – nicht-entziehbare Betriebsmittel – mehrere optionale Teile

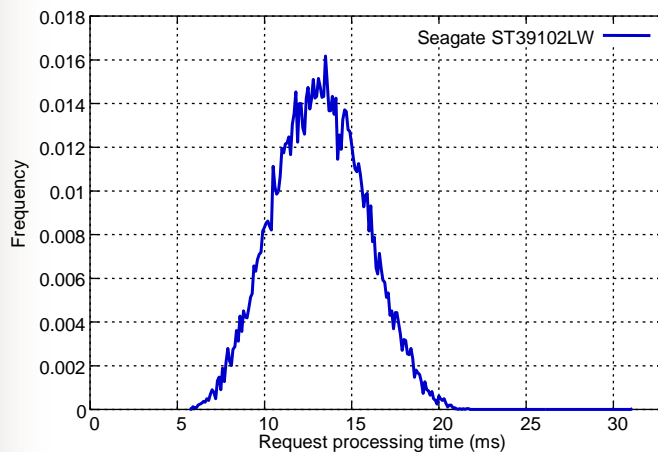
- **Bewertung**

- **Platten-Scheduling**

4 Auftragsströme mit 20 – 10 – 5 – 10 Blöcken zu je 64 KB pro Periode

## Execution Time Distribution

- Disk request processing time  
read requests, 64 KB, random workload



Claude-Joachim Hamann

Quality-Assuring Scheduling

5

## Disk Scheduling: Results

- 4 files, optional parts only, length of period: 500 ms

File	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Requested bandwidth (KB/s)	2560	1280	640	1280
Requested quality $q$	95.0%	90.0%	85.0%	50.0%
Reservation time $r$ (ms)	244.7	112.4	50.0	61.1
Achieved quality $q_{ach}$	94.7%	89.8%	85.0%	50.8%
WCET = 33.2 ms	---	90.0%	85.0%	17.9%

bzw. 75,3% nur für File 1



Claude-Joachim Hamann

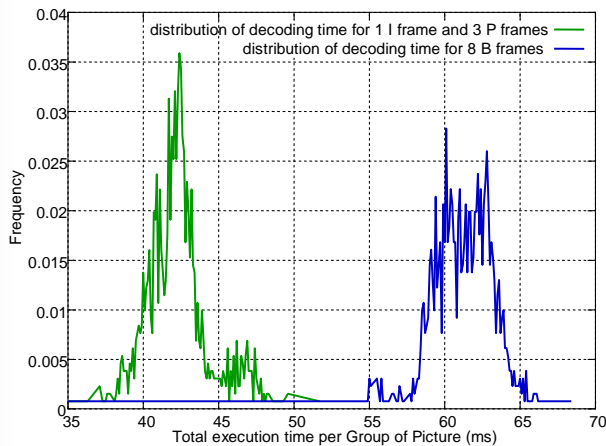
Quality-Assuring Scheduling

6

– CPU-Scheduling

## Experiments vs. Prediction: CPU

- MPEG Group of Picture: IBBPBBPBBPBB  
Execution time distribution



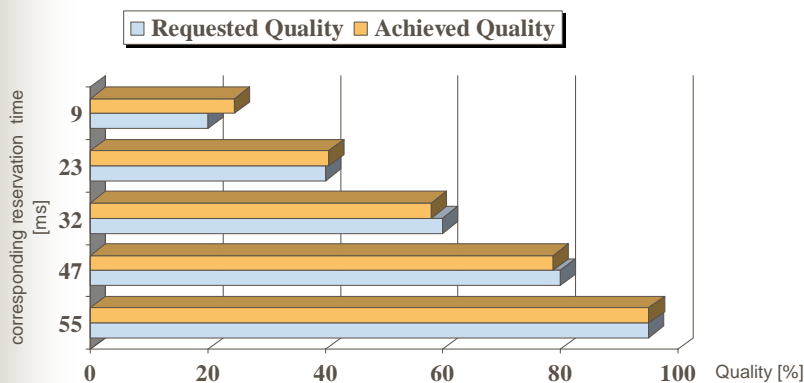
Claude-Joachim Hamann

Quality-Assuring Scheduling

7

## CPU Scheduling: Results

- Length of period: 480 ms
- I and P mandatory, B optional (8 optional parts per period)



Claude-Joachim Hamann

Quality-Assuring Scheduling

8

– Overhead

\* Scheduling: vernachlässigbar!

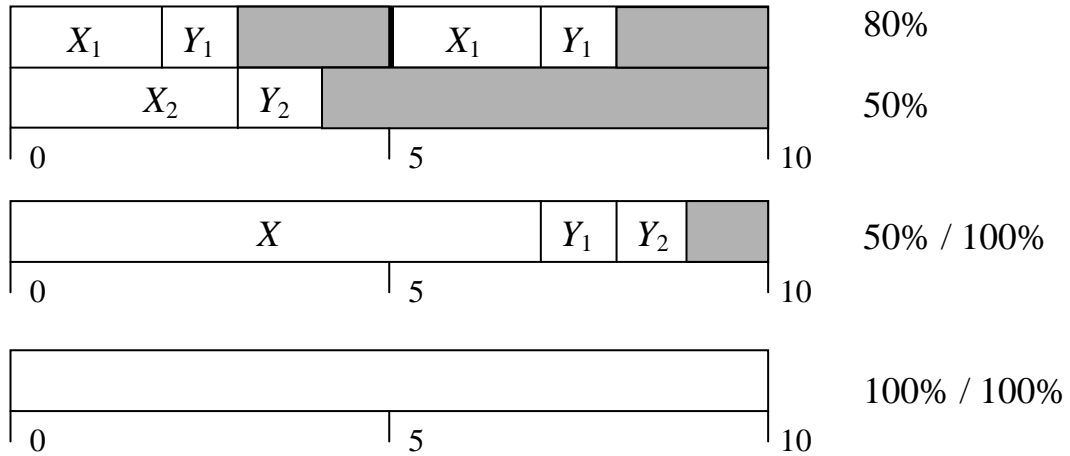
\* Admission: Komplexität  $O(v^2)$   $v$ : Anzahl Werte der Zufallsvariablen

$v$	500	1,000	<b>2,500</b>	5,000	10,000	25,000	50,000
$t_{Adm}/s$	0.015	0.053	<b>0.304</b>	1.204	5.017	75.774	609.191
$q_{ach}$	91.4 %	90.7 %	<b>90.3 %</b>	90.3 %	90.1 %	90.0 %	90.1 %

## 7.4. Harmonische Perioden

- **Prioritätszuordnung**

Weder QMS noch RMS sind optimal.



- **Admission**

QAS

Lars Reuther

Jean Wolter

Claude-J. Hamann

Hermann Härtig

TU Dresden  
Operating  
Systems  
Group

### Harmonic QAS – Admission / Reservation Times

- Admission  $S_2$ :  $d_1$   $d_2$ 

$$\frac{\min(d_1, \sum w_{1i} + \sum r_{1i})}{d_1} + \frac{w_{21}}{d_2} \leq 1$$
- Reservation time  $T_{21}$ :
 
$$r_{21} = \min[ r | P(Y_{21} \leq r \wedge \wedge 3 * \min(d_1, \sum X_{1i} + \sum \min(Y_{1i}, r_{1i})) + X_{21} + Y_{21} \leq d_2) ] \geq q_{21}$$

15

- **Beliebige Perioden**

Simulation:  $r_i \vdash s_i = \min(s \in \mathbb{R} \mid \mathbf{P}(Y_i \leq s_i) \geq q_i) \mid t_i$

Exaktes Modell: Techn. Bericht „Quality-Assuring Scheduling“

[http://os.inf.tu-dresden.de/papers\\_ps/hamann06\\_qas\\_tr3.pdf](http://os.inf.tu-dresden.de/papers_ps/hamann06_qas_tr3.pdf)



# 7.5. QRMS – Quality-Rate-Monotonic Scheduling

- Voraussetzungen und Vorgehen**

Tasks gemäß 7.2

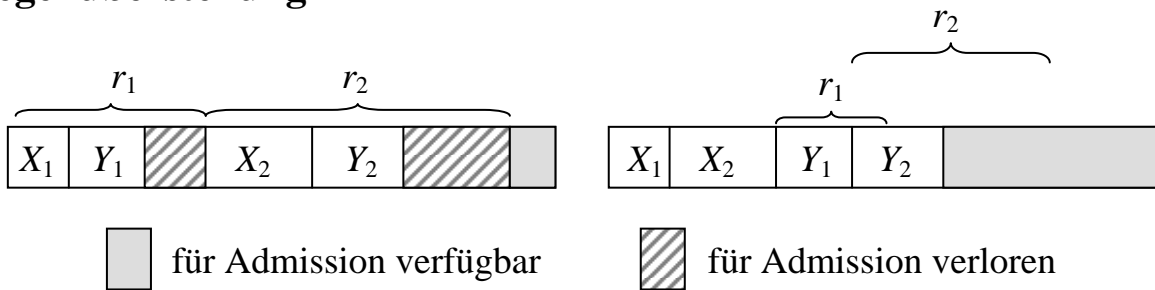
Prioritätszuordnung für *Tasks* gemäß RMS

Reservierungszeit

$$r_i' = \min(r \in \mathbb{R} \mid P(X_i + Y_i \leq r) \geq q_i), \quad r = \max(r_i', w_i)$$

Admission mittels RMS-Kriterien

- Gegenüberstellung**



- Nicht-Optimalität**

QAS  
 Lars Reuther  
 Jean Wolter  
 Claude-J. Hamann  
 Hermann Härtig  
 TU Dresden  
 Operating Systems Group

## QRMS – Quality Rate Monotonic Scheduling

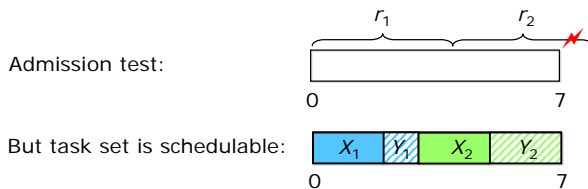
$X_1, Y_1, X_2, Y_2$  identically distributed:

$Z$	1	2
$p$	0.5	0.5

$d_1 = d_2 = 7, \quad q_1 = q_2 = 0.9$

Reservation time:

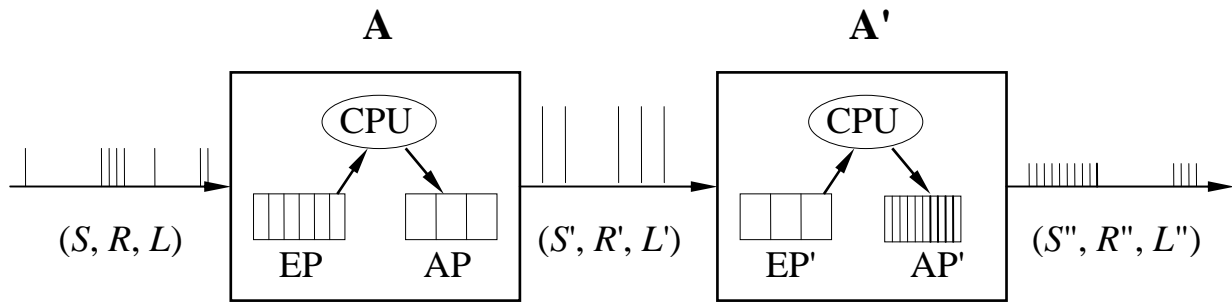
$X_1 + Y_1$	2	3	4	$\Rightarrow r_1' = r_2' = 4$ $\Rightarrow r_1 = r_2 = 4$
$p$	0.25	0.5	0.25	
$\Sigma p$	0.25	0.75	1	



$$P(X_1 + Y_1 + X_2 + Y_2 \leq 7) = 1 - P(X_1 + \dots + Y_2 = 8) = 1 - 1/16 = 0.9275 > 0.9$$

## 7.6. Schwankungsbeschränkte Ströme

- Motivation



$S$ : Spitzenrate

$R$ : mittlere Rate

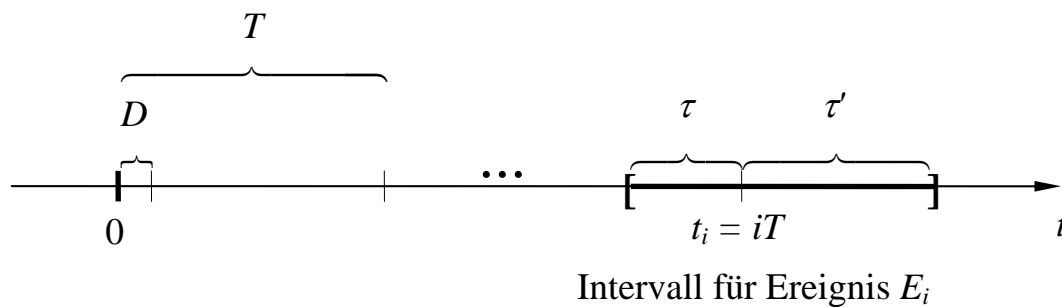
$L$ : maximale Burstlänge

$D$ : Mindestabstand

$T$ : Periodenlänge

$\tau$ : Schwankung

- Schwankungsbeschränkte periodische Ereignisströme



**Definition.** Gegeben seien

$$D, T, \tau, \tau' \in \mathbb{R} \quad \text{mit} \quad T > D > 0, \quad \tau, \tau' \geq 0$$

sowie  $i \in \mathbb{N}$ . Ein  $(\tau, \tau')$ -schwankungsbeschränkter periodischer Ereignisstrom mit konstanter Periode  $T$  und Mindestabstand  $D$  ist dann eine Folge  $(E_i)_{i=0,1,\dots}$  von Ereignissen, die in den Zeitpunkten

$$a_i \in [t_i - \tau, t_i + \tau'] \subseteq \mathbb{R} \quad \text{tatsächliche Ereigniszeit}$$

mit  $t_i = iT$  theoretische Ereigniszeit

eintreten und deren Abstände der Bedingung

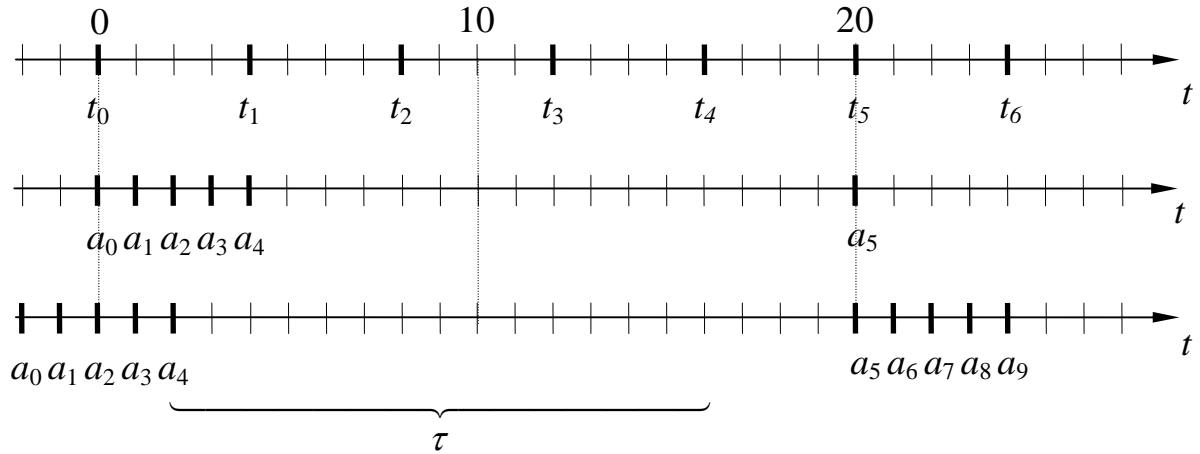
$$a_{i+1} - a_i \geq D \quad \forall i \in \mathbb{N}$$

genügen.

• **Beispiele**

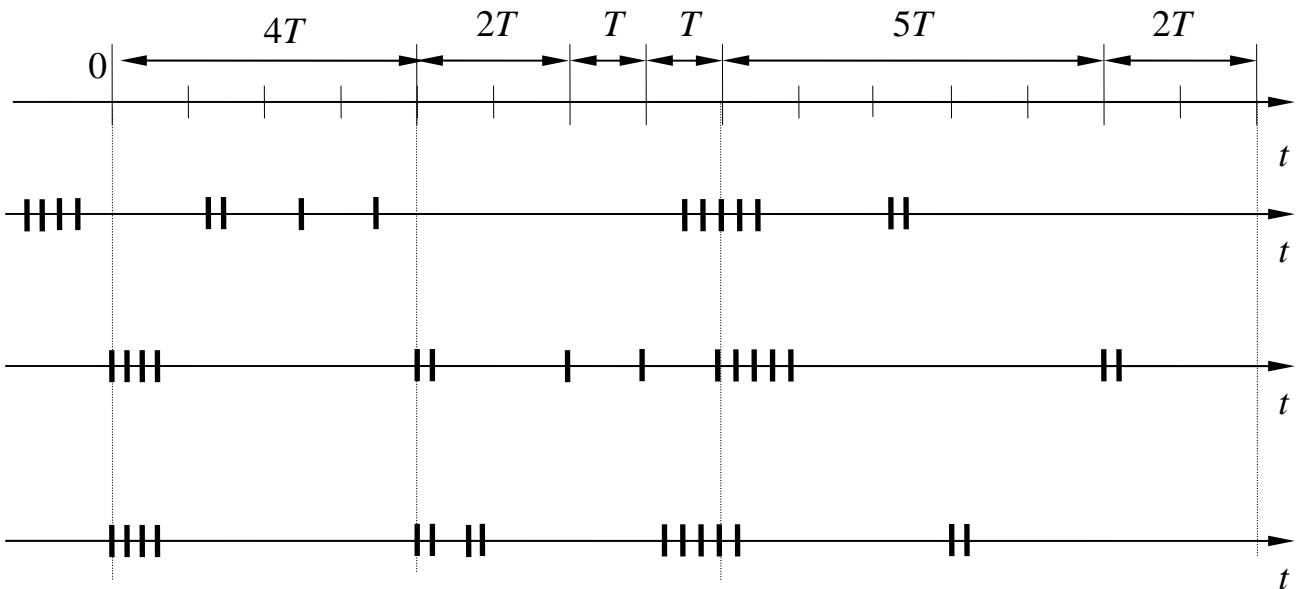
– **Maximaler Burst-Strom** mit  $D = 1$ ,  $T = 4$ ,  $\tau = 14$ ,  $\tau' = 0$ .

Dann ist  $L = 5$ ,  $b_f = -2$ ,  $b_s = 0$ ,  $I_u = 14$ ,  $I_o = 18$ .



– **Dichter Strom von Bursts** der Längen 4, 2, 1, 1, 5, 2

bei  $D = 1$ ,  $T = 4$ ,  $\tau = 14$ ,  $\tau' = 0$ . Dann ist  $L = 5$ .



- **Ausgewählte Ergebnisse**

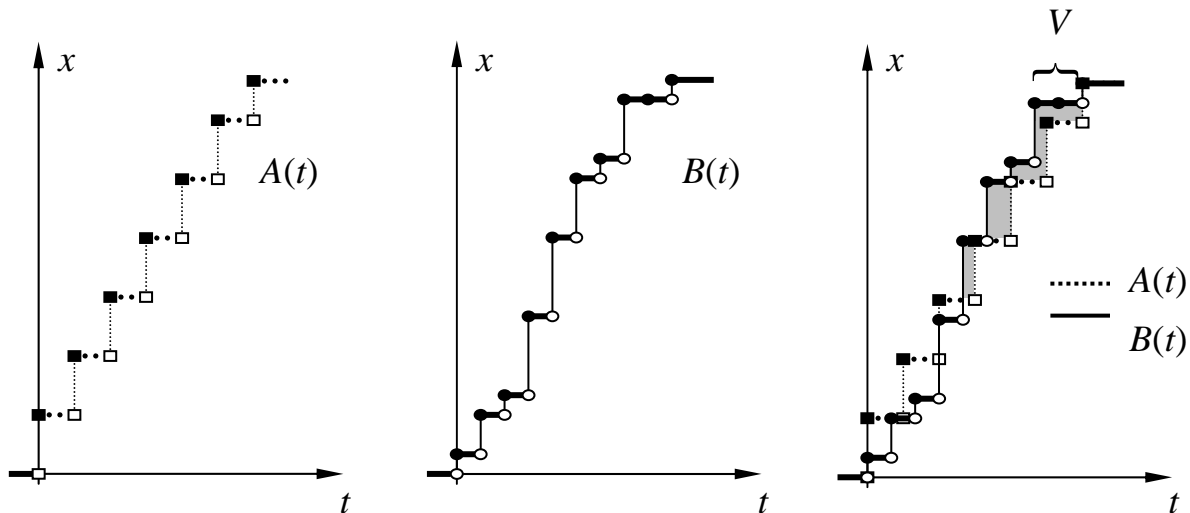
Maximale Burstlänge  $L$      $L = 1 + \left\lfloor \frac{\tau}{T - D} \right\rfloor$

Mindest-Puffergröße  $P$      $P = \left\lceil \frac{\tau}{T} \right\rceil$     bzw.     $P = \left\lceil \frac{(L-1)(T-D)}{T} \right\rceil$

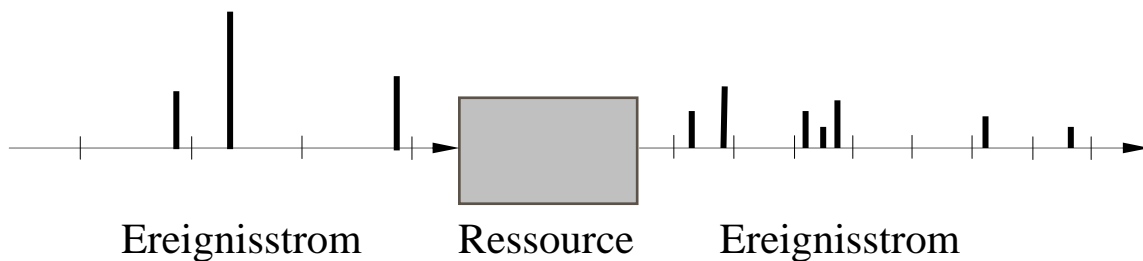
Interburstiness  $I$      $I = \tau + T$

Schwankungsparameter     $\tau \in [(L-1)(T-D), L(T-D))$   
 $\tau \in ((P-1)T, PT]$

**Puffer und Vorlauf**



- **Problem**



HAMANN, CL.-J.; S. ZSHCHALER: Scheduling Real-Time Components Using Jitter-Constrained Streams. Proc. 10<sup>th</sup> IEEE The Enterprise Computing Conference (EDOC), Hong Kong 2006.