7.11 **Dynamisch Prioritäten (Zuordnung nach aufsteigenden Zielzeiten, EDF)**

D7.2 **Definition**

- **Gegeben sei eine Menge von Echtzeitaufträgen und ein Zeitintervall** $[t_1, t_2)$. **Die Rechenlast der Menge im Intervall** $[t_1, t_2)$ **ist**

$$h_{[t_1, t_2)} = \sum_{t_1 \le r_k,\, d_k \le t_2} C_k.$$

- **Gegeben sei eine Menge von Echtzeitaufträgen. Ihr Belastungsfaktor im Intervall [t_1, t_2) ist der Anteil des Intervalls der zur Abarbeitung benötigt wird, d. h.**

$$u_{[t_1, t_2)} = \frac{h_{[t_1, t_2)}}{t_2 - t_1}.$$

- **Der (absolute) Belastungsfaktor ist das Suprenum über alle möglichen Intervalle, d. h.**

$$u = \sup_{0 \le t_1 \le t_2} u_{[t_1, t_2)}$$

BP 2 Prozessorvergabe - vert. Syst.: EDF

S7.8 **Satz (Spuri, 1995)**

Eine Menge von Echtzeitaufträgen ist genau dann ohne Verspätungen abarbeitbar, wenn $u \le 1$ **ist.**

Beweis

a) Notwendig: trivial

b) Hinreichend:

Zum Zeitpunkt t_2 beginne eine Verspätung. Dann ist davor eine Rechenperiode, in der nur Teilaufträge mit kleinerer Zielzeit bearbeitet werden.

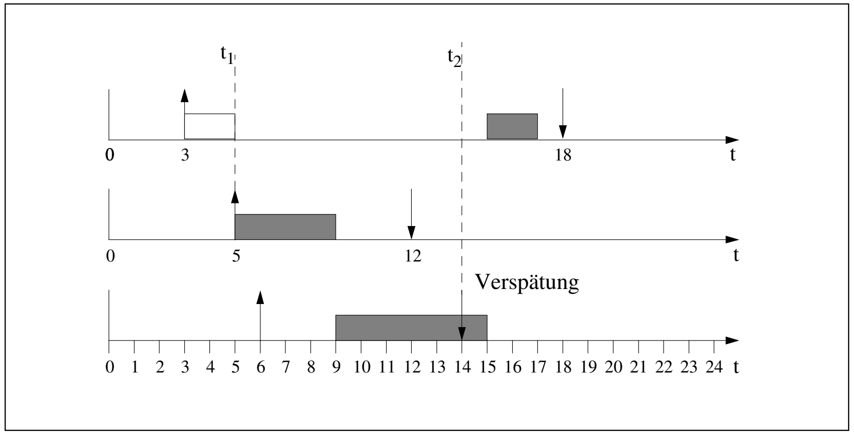
t_1 sei letzter Zeitpunkt vor t_2 derart, daß alle vor t_1 freigegebenen Teilaufträge mit Zielzeiten kleiner oder gleich t_2 abgearbeitet sind.

Dann muß gelten

$$\sum_{t_1 \le r_k,\, d_k \le t_2} C_k > (t_2 - t_1)$$

Also ist $u[t_1, t_2) > 1$ **und somit u > 1, was einen Widerspruch darstellt.**

BP 2 Prozessorvergabe - vert. Syst.: EDF



BP 2 Prozessorvergabe - vert. Syst.: EDF

S7.9 **Satz (Liu and Layland, 1973)**

Jede Menge synchroner periodischer Aufträge mit einer Auslastung $U = \sum_{i=1}^{n} \frac{C_i}{T_i}$

kann genau dann mit EDF ohne Verspätung abgearbeitet werden, wenn $U \le 1$ **ist.**

Beweis

Es genügt zu zeigen, daß der Lastfaktor u gleich U ist.

Nach Satz 7.8 folgt daraus die Aussage.

Für jedes Intervall [t_1, t_2) gilt:

$$\sum_{t_1 \le r_k,\, t_2 \le d_k} C_k \le \sum_{i=1}^{n} \left\lfloor \frac{t_2 - t_1}{T_i} \right\rfloor C_i \le \sum_{i=1}^{n} \frac{t_2 - t_1}{T_i} C_i = (t_2 - t_1) \sum_{i=1}^{n} \frac{C_i}{T_i}$$

also ist $u[t_1, t_2) \le U$.

Sei $t_1 = 0$ and $t_2 = kgV(T_1,...,T_n)$, dann gilt

$$u[t_1, t_2) = \frac{\sum_{i=1}^{n} \frac{t_2}{T_i} C_i}{t_2} = U.$$

Somit ist u = U.

S7.10 Satz (Coffman, 1976)

Jede Menge asynchroner periodischer Aufträge mit einer Auslastung $U = \sum_{i=1}^{n} \frac{C_i}{T_i}$ kann genau dann mit EDF ohne Verspätung abgearbeitet werden, wenn $U \le 1$ ist.

Beweis

Wiederum ist es ausreichend zu zeigen, daß u = U ist.

Für jedes Intervall gilt $[t_1, t_2)$:

$$\sum_{t_1 \le r_k,\, t_2 \le d_k} C_k \le \sum_{i=1}^{n} \left\lfloor \frac{t_2 - t_1}{T_i} \right\rfloor C_i \le \sum_{i=1}^{n} \frac{t_2 - t_1}{T_i} C_i = (t_2 - t_1) \sum_{i=1}^{n} \frac{C_i}{T_i},$$

also ist $u[t_1, t_2) \le U$.

Man betrachte das Intervall [0, s + mH), wobei m eine ganze Zahl größer 0 ist, $s = \max\{s_1,...,s_n\}$, und $H = kgV(T_1,...,T_n)$:

$$h[0, s + mH) \le \sum_{i=1} \frac{mH}{T_i} C_i = mH \sum_{i=1}^{n} \frac{C_i}{T_i} = mHU.$$

Damit folgt $u[0, s + mH) \le \frac{mH}{s + mH} U$

und für beliebiges m $u \ge U$,

also ist u = U.

Mit Satz 7.8 folgt daraus die Aussage.

Daß schon geringfügig komplexere Modelle die Analyse stark erschweren können, zeigt der folgende Satz.

S7.11 Satz (Leung and Merrill, 1980)

Zu entscheiden, ob bei einer synchronen Menge periodischer Aufträge, bei denen die Zielzeiten vor den Periodenenden liegen, keine Verspätungen auftreten ist NP-hart.

7.12 Scheduling in verteilten Echtzeitsystemen

❑ **Besonderheit**

Teilaufträge können mit Empfang einer Nachricht beginnen und deshalb möglicherweise nicht sofort mit Periodenbeginn rechenbereit sein.

Die Differenz aus Periodenbeginn und tatsächlicher Rechenbereitschaft wird als Jitter bezeichnet (ist immer kleiner oder gleich 0!)

Teilaufträge:

- **Ankunftszeitpunkt:** **Zeitpunkt des Periodenbeginns**
- **Freigabezeitpunkt:** **Aufnahme in die Bereit-Warteschlange**

❑ **Es gibt drei wesentliche Untersuchungsmethoden.**

- **Holistische Analyse auf Basis von EDF**

  Statische Zuordnung zu Knoten und lokale Ablaufplanung

- **Spring-Algorithmus**

  Zweigeteilter statischer Algorithmus

  - Welche Aufträge sollen dem gleichen Knoten zugeteilt werden (zur Reduktion des Kommunikationsaufwands)
  - Ermittlung eines Ablaufplans für jeden Knoten

- **Gezielte Weitergabe und Angebotseinholung**

  Angewandt bei dynamischer Erzeugung von Aufträgen, die nicht an einen bestimmten Knoten gebunden sind.

❑ **Genauere Betrachtung der holistischen Analyse**

◆ **Besondere Herausforderung: ’End-zu-End’-Bedingungen**

Das sind Zeitbedingungen die Aktivitäten auf mehreren Knoten umfassen und den Kommunikationsaufwand einbeziehen.

◆ **Einschränkung**

- Empfang höchstens einer Nachricht pro Periode und zwar am Anfang
- ’End-zu-End’-Bedingungen nur zwischen einem Sender und einem Empfänger

◆ **Fünf wesentliche Zeitabschnitte:**

1. **Erzeugungs-Verzögerung: Zeit zum Erzeugen einer Nachricht und zum Einreihen in die Sende-Warteschlange**
2. **Sende-Verzögerung: Zeit von der Einreihung in die Sende-Warteschlange bis zur tatsächlichen Einspeisung in das Verbindungsnetz**
3. **Übertragungs-Verzögerung: Verweilzeit im Netz**
4. **Annahme-Verzögerung: Zeit von der Übergabe durch das Netz bis zur Ankunft beim Empfangsprozeß**
5. **Bearbeitungs-Verzögerung: Zeit für die Bearbeitung der empfangenen Nachricht**

◆ **Globale Betrachtung wird vermieden durch “Attribut-Vererbung”**

- **Nachricht erbt vom Sender Periode und Freigabejitter (Schwankung der Sende-Verzögerung)**
- **Empfänger erbt beide Attribute von der empfangenen Nachricht**

◆ **Gesamtanalyse iterativ:**

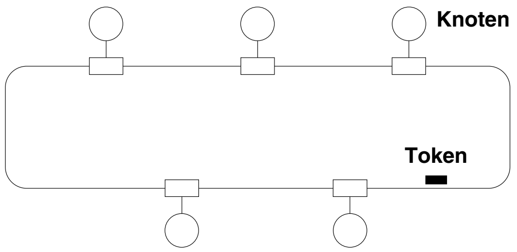
1. **Anfangssituation ohne Jitter**
2. **Lokale Ermittlung der größten Verzögerungen und Nachrichten sowie für das Netz**
3. **Ermittlung der Verzögerungs-Jitter**
4. **Wiederholung ab 2**

Bezeichnungen

- RT_{edf}: Menge aller Anwortzeiten der Aufträge eines Knotens unter Verwendung von EDF
  RT_{net}: Menge der Übertragungsverzögerungen unter Verwendung der Netzstrategie NET
  J: Maximaler Freigabejitter
  P: Netzwerkverzögerungen
- **Iteratives Rechenverfahren**

Teilschritt 1	Teilschritt 2
$RT_1^{(m+1)} = \mathbf{RT}_{edf}(J_1^{(m)})$	$J_1^{(m+1)} = \mathbf{P}_1(RT_{net}^{(m+1)})$
.	.
.	.
.	.
$RT_n^{(m+1)} = \mathbf{RT}_{edf}(J_n^{(m)})$	$J_n^{(m+1)} = \mathbf{P}_n(RT_{net}^{(m+1)})$
$RT_{net}^{(m+1)} = \mathbf{RT}_{net}(J_1^{(m)})$	$J_{net}^{(m+1)} = \mathbf{P}_{net}(RT_1^{(m+1)}, \ldots, RT_n^{(m+1)})$

- **Beispiel: Tokenring**



- Token darf maximale Umlaufzeit TTRT (target token rotation time) nicht überschreiten
- Jeder Knoten p darf das Token für maximal H_p Zeiteinheiten behalten (synchronous bandwidth)
- Der gesamte Overhead des Token-Protokolls pro Umlauf sei τ.

Es muß gelten: $\sum_{p=1}^{n} H_p + \tau \le TTRT$.

S7.12 Wenn bei Verwendung von EDF für ein bestimmtes Ankunftszeitmuster der Teilaufträge eine Verspätung auftritt,

dann existiert in einem Ablauf, in dem die Ankunftszeiten gleichzeitig Freigabezeiten sind,

eine Verspätung ohne vorangehenden Leerstand des Prozessors.

S7.13 Verallgemeinerung des Satzes 7.2 von Liu und Layland

Die längste Antwortzeit eines Auftrags i tritt in einem Intervall mit Auslastungsfaktor 1 auf, in dem jeder andere Auftrag zu Beginn des Intervalls einen Teilauftrag in die Bereitwarteschlange einfügt und seine weiteren Teilaufträge mit maximaler Rate folgen.

Dieser Satz erlaubt eine iterative Berechnung der schlechtesten Anwortzeit gemäß nachfolgender Überlegung.

Es sei in einem Intervall gemäß Satz 7.13 $a \ge -J$ Freigabezeitpunkt des Teilauftrags von Auftrag i, der maximale Antwortzeit aufweist.

Dann besitzt in diesem Intervall der erste Teilauftrag des Auftrags i den

Freigabezeitpunkt $s_i(a) = a + J_i - \left\lfloor \frac{a + J_i}{T_i} \right\rfloor T_i$.

Dementsprechend erzeugt in diesem Interfall Auftrag i die Last $\left(1 + \left\lfloor \frac{a + J_i}{T_i} \right\rfloor\right) C_i$.

Von allen anderen Teilaufträgen haben höchstens $1 + \left\lfloor \frac{a + D_i + J_j - D_j}{T_j} \right\rfloor$ eine Zielzeit

kleiner oder gleich $d = a + D_i$.

Also ist die Last der höherprioren Teilaufträge bezüglich Zielzeit d, die bis zum Zeitpunkt t ankamen

$$W_i(a, t) = \sum_{\substack{j \ne i \\ D_j \le a + D_i + J_j}} \min\left\{\left\lceil \frac{t + J_j}{T_j} \right\rceil, 1 + \left\lfloor \frac{a + D_i + J_j - D_j}{T_j} \right\rfloor\right\} C_j.$$

Die Länge $L_i(a)$ der resultierenden Vollastperiode bezüglich Zielzeit d kann damit iterativ berechnet werden gemäß

$$L_i^{(0)}(a) = \sum_{\substack{j \neq i \\ D_j \leq a + D_i + J_j}} C_j$$

$$L_i^{(m+1)}(a) = W_i(a, L_i^{(m)}(a)) + \left(1 + \left\lfloor \frac{a + J_i}{T_i} \right\rfloor\right) C_i + B_{k(a + D_i)},$$

mit $k(d) = \max\{k : (D_k - J_k \leq d)\}$**, wenn die Aufträge so numeriert sind, daß** $i < j \Rightarrow D_i - J_i \leq D_j - J_j$ **für alle i und j erfüllt ist.**

Es läßt sich zeigen, daß dieses Verfahren nach endlich vielen Schritten konvergiert, wenn $\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$ **ist.**

Die längste Antwortzeit bezogen auf a ist dann $rt_i(a) = \max\{J_i + C_i + B_i, L_i(a) - a\}$.

Man kann zeigen, daß das erste Vollastintervall (unter obigen Annahmen) maximale Länge L besitzt.

Daher müssen nur im Intervall $-J_i \leq a \leq L - J_i - C_i - B_i$ **(also für endlich viele Argumente) die Werte von** $rt_i(a)$ **berechnet und ihr Maximum ermittelt werden.**

- **Mit der gleichen Methode lassen sich wegen der Attributvererbung die Sendeverzögerungen der Kommunikationsprozessoren berechnen.**

  Die auftretenden Verzögerungen beim Netzzugang können als Interferenz eines fiktiven, höherprioren Auftrags betrachtet werden.

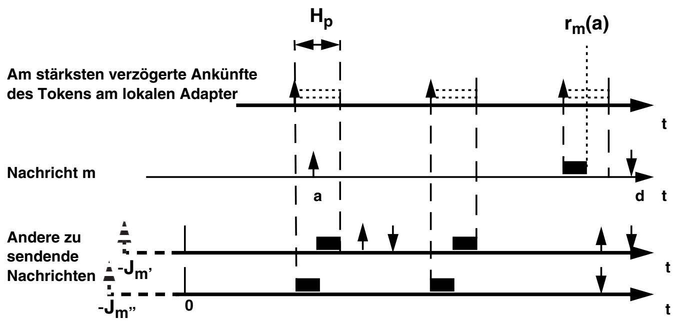
  Dabei sind drei Einflüsse zu berücksichtigen:
  - **Die lokal höherpriore Nachrichtenlast mit Zielzeitpunkten kleiner** $a + D_i$.
  - **Bisherige Nachrichtenpakete zur Übermittlung der Gesamtnachricht.**
  - **Der Einfluß anderer Knoten auf den Netzzugang.**

  - **Die ersten beiden sind gut abschätzbar; der dritte bedarf einer gesonderten Überlegung.**

- **Sendeverzögerung bei synchroner Nutzung des Tokenrings**
  - **Nachrichten werden in Pakete zerlegt**
    - **Cm** — **Zahl der Pakete für Nachricht m**
    - **nm** — **Auftrag i versendet Nachricht m in jeder nm-ten Periode**

BP 2 Prozessorvergabe - vert. Syst.: EDF

- **Mögliches Szenario**



a	Ankunft der Nachricht m in der Sendewarteschlange
H_p	Intervall, in dem der Knoten p Packete senden darf

BP 2 Prozessorvergabe - vert. Syst.: EDF

ρ	Zeit zur Einspeisung einer Nachricht
$\mathbf{P}$	Verweilzeit im Netzwerk
$L_{m,\,k}(a)$	Zeit, die von höherprioren Paketen verbraucht wird

- $$r_m = \max_{a \geq -J_m}\{r_m(a)\}$$
- **Verzögerung der ersten k Pakete**

$$r_{m,\,k}(a) = \max\{J_m + B_m + k\rho + \mathbf{P},\ L_{m,\,k}(a) + \rho + \mathbf{P} - a\}$$

Die für höherpriore Nachrichten verbrauchte Zeit $L_{m,\,k}(a)$ wird bestimmt durch

1. zu sendende Nachrichten mit Zielzeit kleiner oder gleich $a + D_m$
2. vorangehende Pakete
3. Zeit für Umlauf des Tokens

Für alle drei Anteile sind die Obergrenzen gut abschätzbar

BP 2 Prozessorvergabe - vert. Syst.: EDF

- **Bestimmung des Wertes für a, der zur maximalen Verzögerung führt mit gleicher Methode wie bei Ermittlung der gleichen Methode, wie sie zur Ermittlung der Erzeugungsverzögerung angewandt wurde.**

❑ **Annahmeverzögerung**
- Wird für jedes Zeitintervall bestimmt durch die Zahl ankommender Nachrichten
- Kann anhand der Modellannahmen gut abgeschätzt werden

❑ **Bearbeitungsverzögerung**
- Berechnung wie bei Erzeugungsverzögerung, wobei die Annahmeverzögerung als Last höchster Priorität behandelt wird.

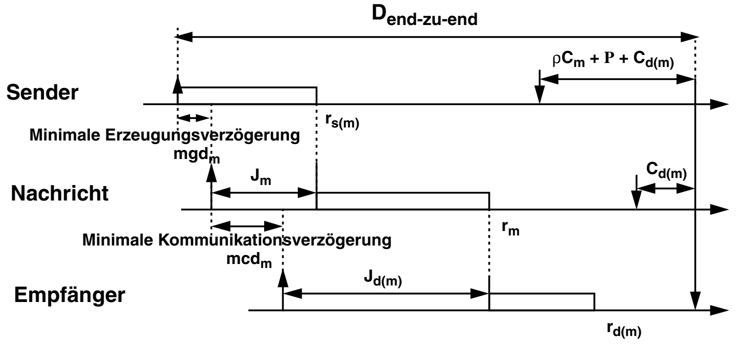
BP 2 Prozessorvergabe - vert. Syst.: EDF

❑ **Bestimmung der Zielzeitenvorgabe für die einzelnen Komponenten aus der End-zu-End-Verzögerung**

- **Optimale Zeilzeitenvorgabe ungelöst**

Prozessorvergabe - vert. Syst.: EDF

- **Approximation**



- $D_{d(m)} = \min\{D_{d(m)}, D_{end-zu-end} - mgd_m - mcd_m\}$
- $D_m = \min\{D_m, D_{d(m)} - C_{d(m)} + mcd_m\}$
- $D_{s(m)} = \min\{D_{s(m)}, D_m - mcd_m + mgd_m\}$