

# **Kapitel 10**      **“Traffic Engineering“**

<b>10.1</b>	<b>Teilgebiete des "Traffic Engineering"</b>	<b>2</b>
<b>10.2</b>	<b>Formale Lastbeschreibung und Lastcharakterisierung</b>	<b>15</b>
<b>10.3</b>	<b>Lastmessungen in Rechnernetzen und bei Medienkommunikation</b>	<b>24</b>
<b>10.4</b>	<b>Lastmodellierung für Rechnernetzbenutzer</b>	<b>29</b>

# 10 “TRAFFIC ENGINEERING“

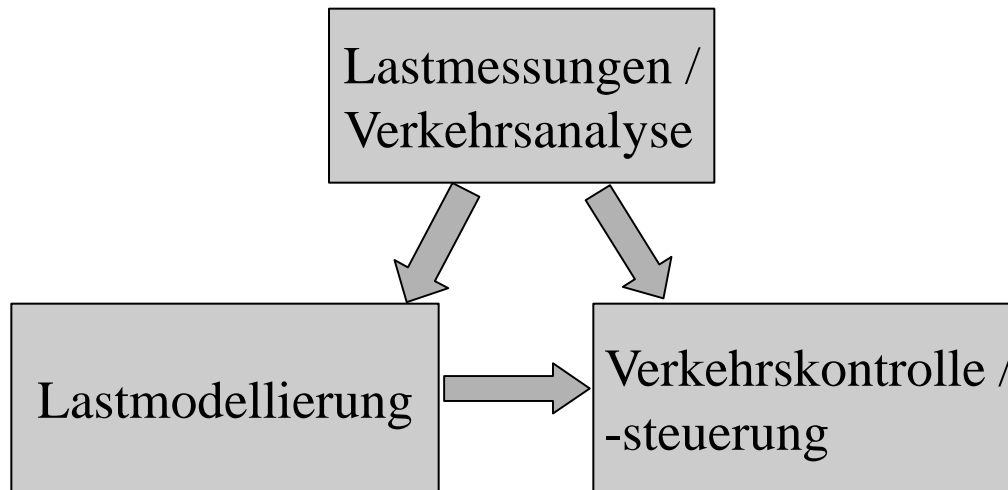
## 10.1 Teilgebiete des “Traffic Engineering“

Def. “**Traffic Engineering**“ :

Technisch-wissenschaftliche Beschäftigung mit Verkehr in Kommunikations-/  
Rechnernetzen.    



Teilgebiete des “Traffic Engineering“:



*Nota bene:* Analogie zum Straßenverkehr

## Nutzen des “Traffic Engineering” am Beispiel des Betriebs eines Unternehmensnetzes ?

⇒ Unterstützung des

- **Leistungsmanagements:**  
Engpassermittlung, “Tuning”, Netzverhalten bei steigender Last
- **Dienstgüte-(QoS-)Managements:**  
Ermittlung repräsentativen Benutzerverhaltens zur Prognose zukünftiger Betriebsmittelbedarfe
- **Abrechnungsmanagements:**  
Erfassung individueller Verkehrsströme zur Charakterisierung des Betriebsmittelverbrauchs einzelner Benutzer
- **Konfigurierungsmanagements:**  
künstliche Verkehrsgeneratoren, realitätsnahe Netzmodellierungen dank realistischer Lastcharakterisierungen
- **Sicherheitsmanagements:**  
Verkehrsanalysen als Basis für die Erkennung von Angriffsversuchen  
(*aber* : Charakterisierung von Benutzerverhalten auch selbst ein Sicherheitsproblem)

... und zur weiteren Motivation für  
***Traffic Engineering im Kontext des Internet***

**Meldung vom 26.01.2011 10:27 Uhr**

***Internet-Verkehr soll sich bis 2015 verzwanzigfachen***

Frankfurt/Berlin (dpa) - Der Datenverkehr im Netz wird sich bis 2015 verzwanzigfachen - dies erwarten die Betreiber des zentralen Internet-Knotens (DE-CIX) in Frankfurt am Main. Inzwischen schleust der Netzknoten jede Sekunde 1,4 Terabit an Daten durch, also 1,4 Billionen Bit, wie die DE-CIX Management GmbH am Mittwoch mitteilte. Im Januar 2010 waren es erst 700 Gigabit gewesen.



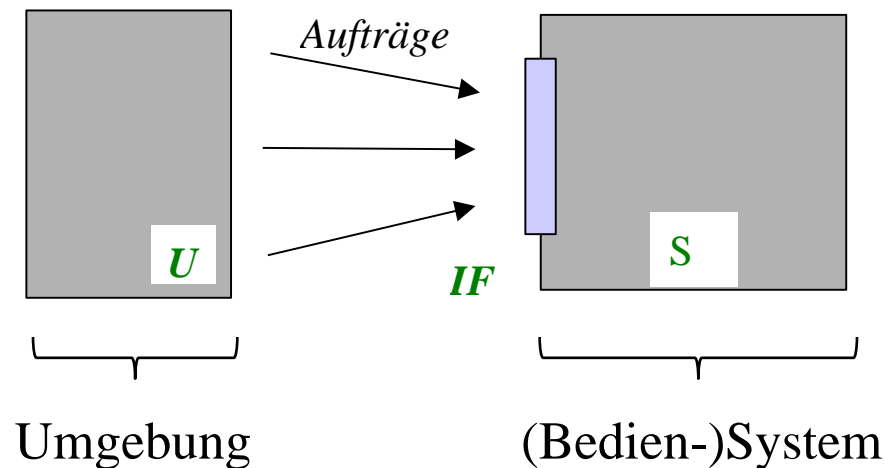
**Def. (Angebotene) Last**  $L = L(U, S, IF, T)$  :  
engl. “(offered) load”

Eine **Last**  $L$  sei definiert als (vollständige) **Sequenz von Aufträgen**, die

- durch eine **Umgebung**  $U$
- an ein **Bediensystem**  $S$
- über eine **wohldefinierte Schnittstelle**  $IF$
- während eines **Beobachtungsintervalls**  $T$

übergeben werden.

Wir bezeichnen  $L$  als die seitens  $U$  für  $S$  während  $T$  generierte Last.



# Zur Lastdefinition

→ zu beachten :

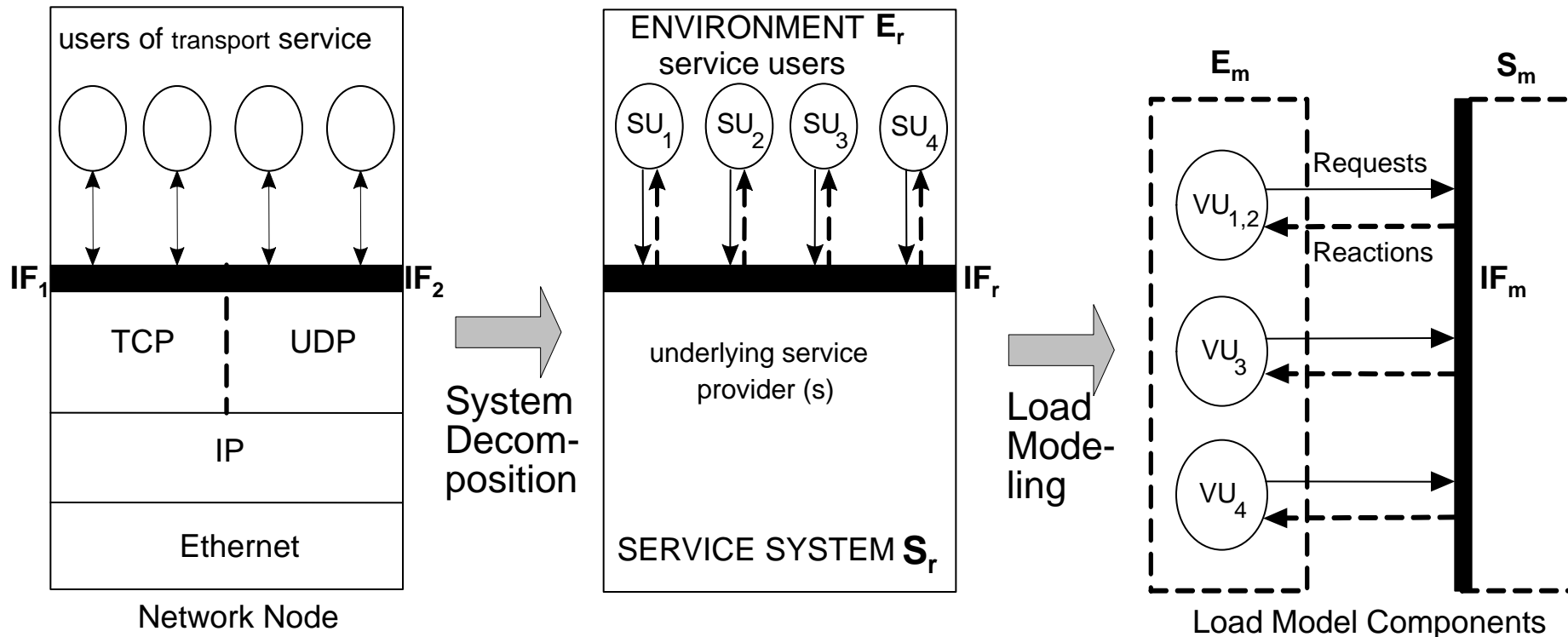
- Zerlegung des Gesamtsystems in **S** und **U** dergestalt, dass wohldefinierte, beobachtbare Schnittstelle (**IF**) resultiert;  
*nota bene* : **IF** evtl. geographisch verteilt realisiert;
- Beispiele für auftragsgenerierende Umgebungen:  
Menschen, Hardwarekomponenten (z.B. Kamera),  
System- bzw. Anwendungsprozesse (z.B. Multimedia-Anwendungen);
- Ziel der Studie und gewähltes **IF** bestimmen Aufträge;
- starker Einfluss von **T** auf **L**;
- $L = L(U, S, IF, T)$  ist **nicht** dasselbe wie  $L = (U, S, IF, T)$  !!!

Beispiele für **IF**:

- Dienstschnittstellen in typischen Protokollhierarchien
- "Socket-Schnittstelle" von TCP
- $S_0$  -/ $S_{2m}$  - Schnittstelle bei ISDN
- virtuelle Verbindungen (VCs) bei ATM-Netzen

# Veranschaulichung unserer Definition von Last sowie des Procederes bei Lastmodellierung

*Last*



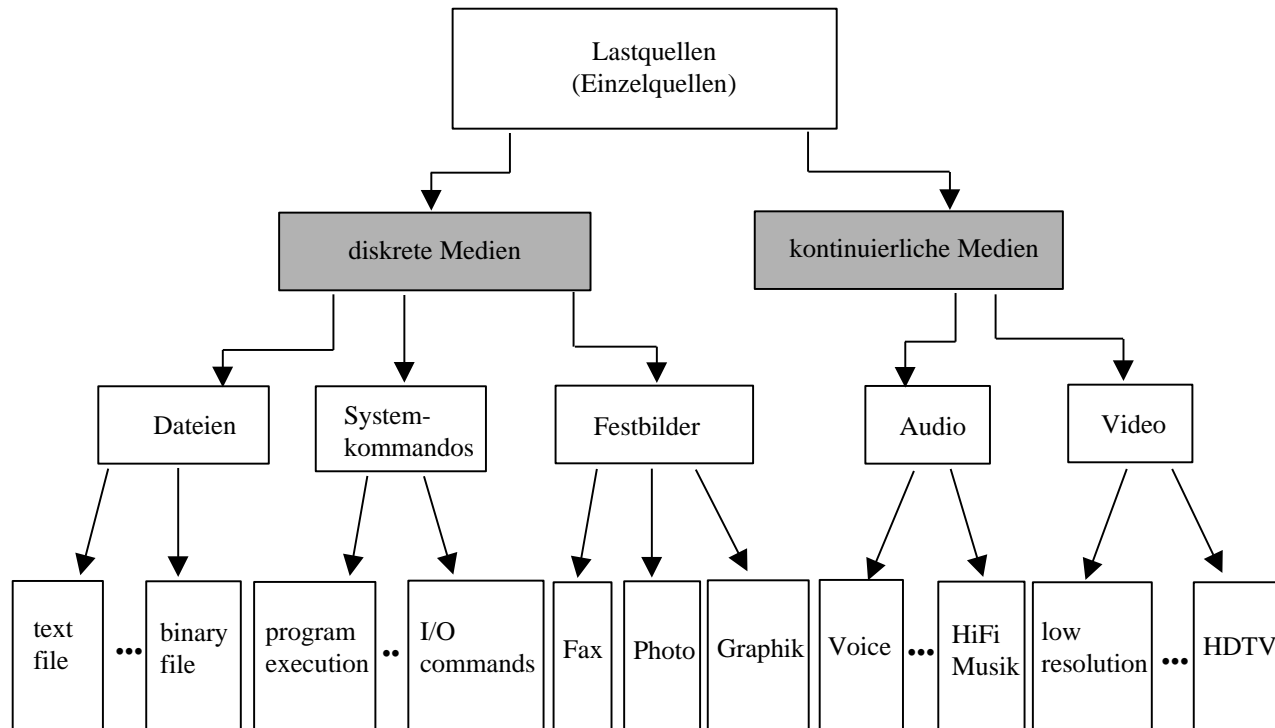
*Nota bene:* Index **r** for real system and index **m** for model (domain)

# Lastquellen

## Einzelquellen (Elementarquellen) *versus* komplexe Lastquellen

- **Einzelquellen** : elementarer Auftrag bzw. Auftragsfolge an IF
  - Beispiele : Folge von Dienstprimitiven bei Dateitransfer, Übertragung einer Folge von PCM-codierten Abtastwerten (z.B. im Rahmen einer Bildtelefonieanwendung).
- **Komplexe Lastquellen** : z.B. modellierbar als Überlagerung von Einzellastquellen.

⇒ **Klassifikation** von elementaren Lastquellen :





## *Diskrete versus kontinuierliche Medien*

➤ ad **diskrete Medien** (*discrete media*) :

evtl. relativ unterschiedliche Typen von Aufträgen mit unregelmäßigen Ankunftszeitpunkten an IF

➤ ad **kontinuierliche Medien** (*continuous media*) :

*auch:* **isochroner Verkehr** →

über größere Zeitintervalle, z.B. min-/h-Bereich, (näherungsweise) konstante Zwischenankunftszeiten und zumeist sehr ähnliche Typen von Aufträgen (Beispiel : zu übertragende Videoframes)

*nota bene:* „Kontinuierlich“ bedeutet somit NICHT : „kontinuierlich im math. Sinne“, sondern eher „periodische Auftragsgenerierung“

## Motivation für “Traffic Engineering“

### ➤ Nutzen einer **Verkehrsbeeinflussung** u.a.

- zur Erzeugung von "sympathischem Verkehr" für Netz  
→ Verkehrsglättung, ratenbasierte Zugangskontrolle für Aufträge, ...
- zur Unterstützung der Netzsicherheit durch Filtern von Aufträgen

→ vgl. Kap. 8 zu Echtzeit-/Medienkommunikation (insbesondere die Beispiele "*Smoothing*", "*Token Bucket*" bzw. "*Leaky Bucket*")

### ➤ Nutzen einer **Lastmessung** u.a.

- zur Protokollierung des Benutzerverhaltens  
→ Nutzung der Messdaten (ggf. nach statistischer Auswertung) als Basis für Verhaltensprognosen, für Erkennung "repräsentativen" Benutzerverhaltens, ...
- als Ausgangspunkt für eine realistische Lastmodellierung
- als Basis für die Erzeugung hinreichend valider Lastgeneratoren

## Motivation für “*Traffic Engineering*“ (Forts.)

- Nutzen einer **Lastmodellierung** u.a.
  - zur Gewinnung und Parametrisierung von hinreichend realitätsnahen Lastmodellen für analytische oder simulative Kommunikations- und Rechnernetzmodelle
  - zur Gewinnung von künstlichen Lastgeneratoren für existierende Rechnernetze
  
- Nutzen einer **Lastspezifikation** bzw. einer formalen Lastbeschreibung u.a.
  - zur exakten Beschreibung von Lasten bei Messung und Modellierung
  - zur präzisen Spezifikation des Verhaltens von Lastgeneratoren (in Modellen oder in existierenden Rechnernetzen).

# Zur Weiterentwicklung des Internet-Verkehrs I.

DKR: X.12

[laut **Cisco-Studie** “Global IP traffic forecast and methodology; Approaching the Zettabyte<sup>\*)</sup> Era“; White paper series (2009); *siehe: [www.cisco.com](http://www.cisco.com)*]

- bis 2013 *verdoppelt sich IP-Verkehr global alle 2 Jahre* nahezu;
- derzeitig 2-stelliger %-Anteil von *Peer-to-Peer (P2P-)Verkehr schrumpft* im Verhältnis zum gesamten IP-Verkehr, wächst in seinem Volumen jedoch weiterhin;
- *Videodatenströme* (resultierend aus unterschiedlichen Anwendungen wie IPTV, Video-on-Demand/VoD, etc) werden voraussichtlich bis 2013 [in Prognose aus 2011, s.u., ist indes hier vom Jahr 2015 die Rede !] den IP-Verkehr mit einem *Anteil von mehr als 90 %* dominieren;
- *Sprachdatenströme* (insbes. resultierend aus VoIP-Anwendungen) werden voraussichtlich bis 2013 nur noch einen sehr geringen *Anteil < 1 %* an dem gesamten IP-Verkehr ausmachen – trotz weiterhin sehr häufiger Nutzung von Sprachdiensten im Internet.

<sup>\*)</sup> Zettabyte (kurz: ZB) =  $10^{21}$  Byte = 1,000,000,000,000,000,000,000 Byte =  $1000^7$   
[0.75 ZB als Prognose für 2014 für das jährliche globale Verkehrsaufkommen im Internet; ca. 2 Jahre benötigt, um die im Jahr 2014 pro sec übertragenen Videos anzuschauen bzw. 72 Mio. Jahre für die jährl. übertragenen Videos, vgl. [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-481360\\_ns827\\_Networking\\_Solutions\\_White\\_Paper.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html) ]

*Nota bene:* Intensive Forschungsaktivitäten zu IP-Verkehr bei TKRN !

# Zur Weiterentwicklung des Internet-Verkehrs II.

DKR: X.13

[laut **Cisco-Studie** “Global IP traffic forecast and methodology;

**Entering** the Zettabyte Era“; White paper series (June 1, 2011); *siehe: [www.cisco.com](http://www.cisco.com)*]

*June 1, 2011* : This forecast is part of the Cisco® Visual Networking Index (VNI), an ongoing initiative to track and forecast the impact of visual networking applications. This document presents the details of the Cisco VNI global IP traffic forecast and the methodology behind it.

## **Executive Summary**

- **Annual global IP traffic will reach the zettabyte threshold (966 exabytes or nearly 1 zettabyte) by the end of 2015.** In 2015, global IP traffic will reach 966 exabytes per year or 80.5 exabytes per month.
- **Global IP traffic has increased eightfold over the past 5 years, and will increase fourfold over the next 5 years.** Overall, IP traffic will grow at a compound annual growth rate (CAGR) of 32 percent from 2010 to 2015.
- **In 2015, the gigabyte equivalent of all movies ever made will cross global IP networks every 5 minutes.** Global IP networks will deliver 7.3 petabytes every 5 minutes in 2015.
- **The "terabyte club" will reach 6 million by 2015.** In 2015, there will be 6 million Internet households worldwide generating over a terabyte per month in Internet traffic, up from just a few hundred thousand in 2010. There will be over 20 million households generating half a terabyte per month in 2015.
- **The number of devices connected to IP networks will be twice as high as the global population in 2015.** There will be two networked devices per capita in 2015, up from one networked device per capita in 2010. Driven in part by the increase in devices and the capabilities of those devices, IP traffic per capita will reach 11 gigabytes per capita in 2015, up from 3 gigabytes per capita in 2010.
- **A growing amount of Internet traffic is originating with non-PC devices.** In 2010, only 3 percent of Internet traffic originated with non-PC devices, but by 2015 the non-PC share of Internet traffic will grow to 15 percent. PC-originated traffic will grow at a CAGR of 33 percent, while TVs, tablets, smartphones, and machine-to-machine (M2M) modules will have growth rates of 101 percent, 216 percent, 144 percent, and 258 percent, respectively.
- **Traffic from wireless devices will exceed traffic from wired devices by 2015.** In 2015, wired devices will account for 46 percent of IP traffic, while Wi-Fi and mobile devices will account for 54 percent of IP traffic. In 2010, wired devices accounted for the majority of IP traffic at 63 percent.
- **Busy-hour traffic is growing more rapidly than average traffic.** Busy-hour traffic will increase fivefold by 2015, while average traffic will increase fourfold. During an average hour in 2015, the traffic will be equivalent to 200 million people streaming high-definition video continuously. During the busy hour in 2015, the traffic will be equivalent to 500 million people streaming high-definition video continuously.

# Zur Weiterentwicklung des Internet-Verkehrs III.

DKR: X.14

[laut **Cisco-Studie** “Global IP traffic forecast and methodology; **Entering** the Zettabyte Era“; White paper series (June 1, 2011); *siehe: [www.cisco.com](http://www.cisco.com)*]

*June 1, 2011*

This forecast is part of the Cisco® Visual Networking Index (VNI), an ongoing initiative to track and forecast the impact of visual networking applications. This document presents the details of the Cisco VNI global IP traffic forecast and the methodology behind it.

## Video Highlights

- **Global Internet video traffic surpassed global peer-to-peer (P2P) traffic in 2010, and by 2012 Internet video will account for over 50 percent of consumer Internet traffic.** As anticipated, as of 2010 P2P traffic is no longer the largest Internet traffic type, for the first time in 10 years. Internet video was 40 percent of consumer Internet traffic in 2010 and will reach 50 percent by year-end 2012.
- **It would take over 5 years to watch the amount of video that will cross global IP networks every second in 2015.** Every second, 1 million minutes of video content will cross the network in 2015.
- **Internet video is now 40 percent of consumer Internet traffic, and will reach 62 percent by the end of 2015,** not including the amount of video exchanged through P2P file sharing. The sum of all forms of video (TV, video on demand [VoD], Internet, and P2P) will continue to be approximately 90 percent of global consumer traffic by 2015.
- **Internet video to TV tripled in 2010.** Internet video to TV will continue to grow at a rapid pace, increasing 17-fold by 2015. Internet video to TV will be over 16 percent of consumer Internet video traffic in 2015, up from 7 percent in 2010.
- **Video-on-demand traffic will triple by 2015.** The amount of VoD traffic in 2015 will be equivalent to 3 billion DVDs per month.
- **High-definition video-on-demand will surpass standard definition by the end of 2011.** By 2015, high-definition Internet video will comprise 77 percent of VoD.

# 10.2 Formale Lastbeschreibung und Lastcharakterisierung

## ➤ Aspekte einer Lastcharakterisierung

- **Charakterisierung des Ankunftsprozesses** der Aufträge  
(an betrachteter Schnittstelle IF während Zeitintervall T)
- **Charakterisierung der einzelnen Aufträge** selbst  
→ z.B. Charakterisierung über Auftragstypen und Auftragsattribute



## ➤ Anforderungen an eine formale Lastbeschreibungstechnik :

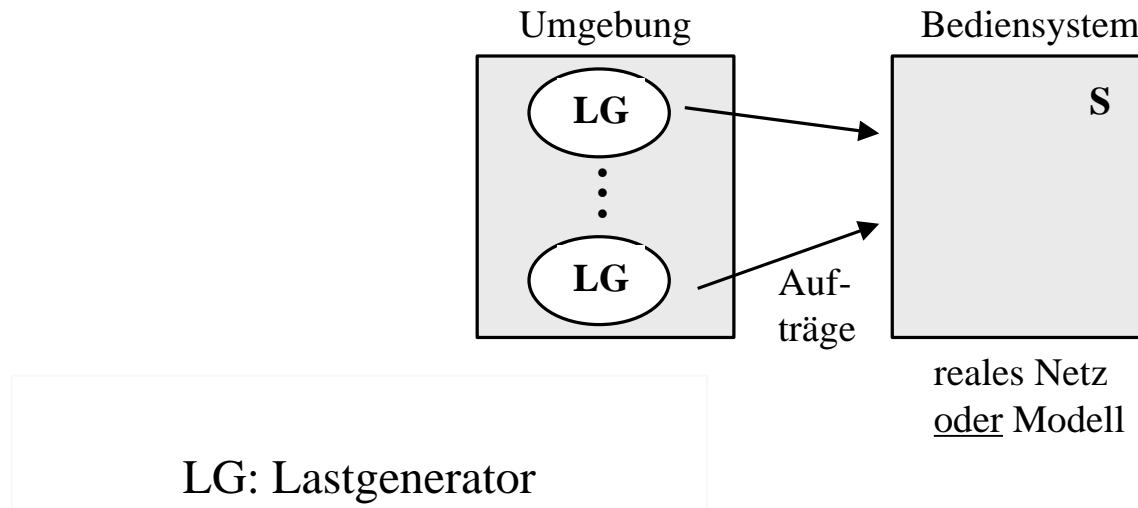
- **allgemein :**  
präzise Spezifikation sowohl der einzelnen Aufträge als auch des Auftragsankunftsprozesses
- **speziell :**
  - klare Trennung der Spezifikation der zeitlichen Aspekte (Ankunftsprozess) von der Spezifikation der entstehenden Aufträge (Realsystem- oder Modellaufträge)
  - Beschreibungstechnik nutzbar sowohl für deterministische sowie für probabilistische Ankunftsprozesse (z.B. Festlegung mittels Trace oder Ankunftszeitverteilungen)
  - möglichst einfache Verwendbarkeit und Interpretierbarkeit der Lastbeschreibungstechnik für ihre(n) BenutzerIn
  - Unterstützung einer schrittweisen Vergrößerung oder Verfeinerung einer vorliegenden Lastbeschreibung u.ä.

# Beispiel einer formalen Lastbeschreibungstechnik

Zu Details vgl. u.a.

- B.E. Wolfinger : *Characterization of Mixed Traffic Load in Service-Integrated Networks*. Systems Science Journal, Vol. 25, No.2 (1999), 65-86  
sowie
- J. Cong : *Load Specification and Load Generation for Multimedia Traffic Load in Computer Networks*. Dissertation, Department Informatik, Universität Hamburg, 2006, erschienen in: Wolfinger B.E. (Hrsg.), Berichte aus dem Forschungsschwerpunkt Telekommunikation und Rechnernetze, Band 5, Shaker-Verlag, Aachen, 2006

## Allgemeine Sicht :

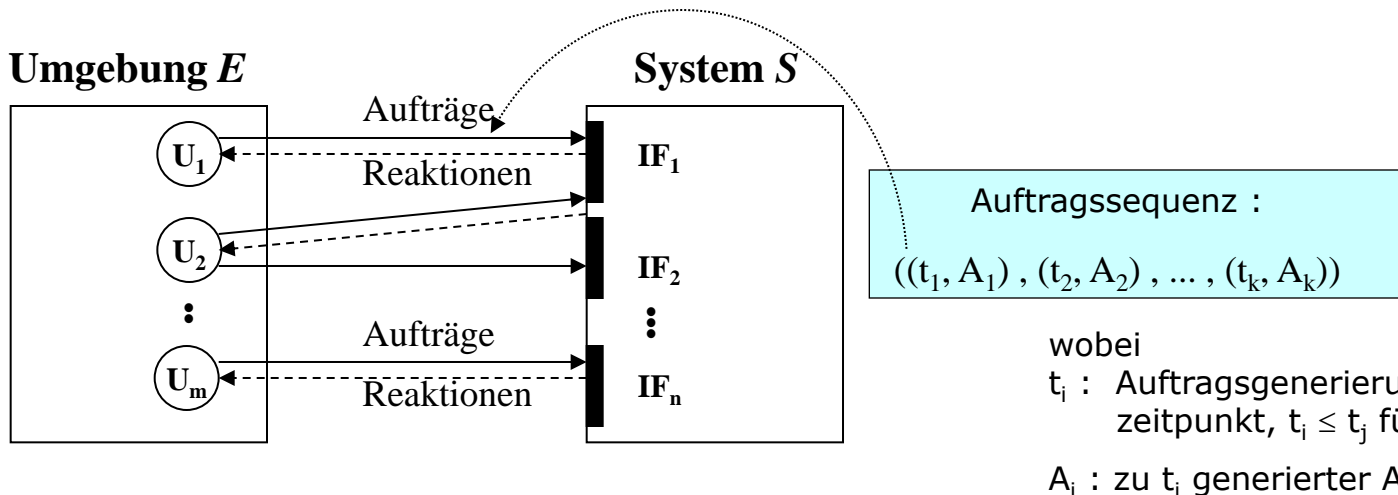


*Nota bene:* Lastgenerator generiert Auftragssequenz wie ein „Benutzer“ des entsprechenden Dienstes an der gegebenen (Dienst-)Schnittstelle; *hier angestrebt:* Formale Beschreibung der Auftragssequenz



# Lasten in Kommunikationsnetzen

- **Umgebung** (*environment*) **E**: erzeugt (Sequenzen von) Aufträgen, die über wohldefinierte Schnittstelle (*interface*, z.B. Dienstschnittstelle) IF übergeben  
→ z.B. E = Menge der Benutzer eines Netzdienstes
- (Bedien-) **System** **S**: Bearbeitet die Aufträge, z.B. zu übertragende Dateneinheiten; Reaktionen (z.B. Bereitstellung von Resultaten einer Auftragsbearbeitung) beobachtbar seitens E  
→ z.B. S = Menge der Erbringer eines (evtl. verteilt realisierten) Netzdienstes



*Nota bene:* Ein **Auftrag** an einer Schnittstelle **IF** wäre z.B. ein Verbindungsaufbauwunsch oder eine Beanspruchung eines Dienstes zur Übertragung von Nutzdaten (die zu übertragenden Dateneinheiten selbst wären indes kein Auftrag, sondern ggf. ein Auftragsattribut für einen Sendeauftrag)

⇒ **TEILZIELE bei einer Lastbeschreibung:**

(G<sub>1</sub>) Spezifikation der einzelnen Aufträge

(G<sub>2</sub>) Spezifikation des Lastgenerierungsprozesses über der Zeit

- **(G<sub>1</sub>) Spezifikation eines einzelnen Auftrags**

- **Genereller Ansatz:**

```

REQUEST_TYPE    {eindeutiger Typ}
begin
  attribute1 : domain of attribute1;
  attribute2 : domain of attribute2;
  ⋮
  attributek : domain of attributek;
end {des Auftragsstyps REQUEST_TYPE}.
  
```

- **Spezielles Beispiel** eines Datentransfer-Auftrags :

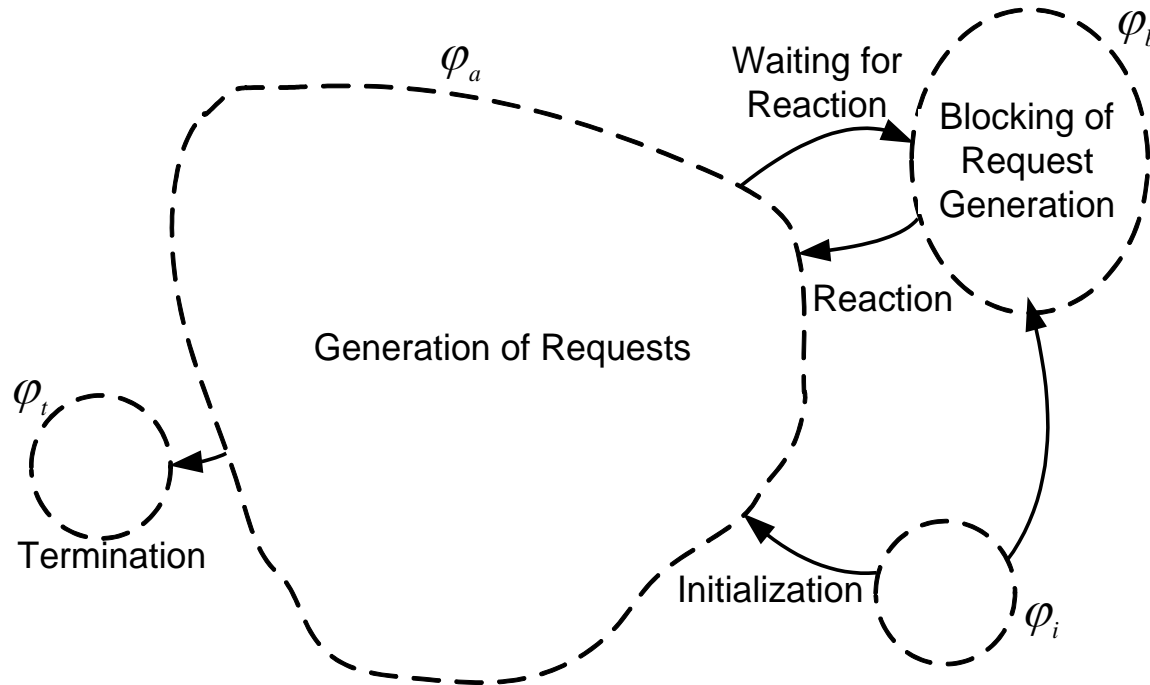
```

SEND_PDU_request
begin
  VIRTUAL_CIRCUIT_ID : integer;
  LENGTH : integer;
end {der SEND_PDU-Attribute}.
  
```



# (G<sub>2</sub>) Spezifikation des Lastgenerierungsprozesses über der Zeit

## Spezifikation des “Benutzer“- (Umgebungs-) Verhaltens bezüglich Auftragsgenerierung



$$\Phi = \{\varphi_i, \varphi_a, \varphi_b, \varphi_t\}$$

**Def. Benutzerverhaltensautomat (User Behaviour Automaton)**  $U = \{\varphi_i, \varphi_a, \varphi_b, \varphi_t, T_\varphi\}$  repräsentiert durch erweiterten, endlichen Automaten basierend auf einer Menge von Makrozuständen  $\Phi = \{\varphi_i, \varphi_a, \varphi_b, \varphi_t\}$ ,  $\varphi_i$ : idle-Zustand,  $\varphi_a$ : active-Zustand,  $\varphi_b$ : blocked-Zustand,  $\varphi_t$ : termination-Zustand,  $T_\varphi$ : Menge von Transitionen zwischen Makrozust.

# Verfeinerung von Makrozuständen

➤  $\varphi_a$ :

D-Zustände  $D_{ij}$ : (benutzerspezifische) Verzögerungen zwischen der Erzeugung aufeinanderfolgender Aufträge innerhalb einer Auftragssequenz  
 → z.B. Verzögerungen gemäß Verteilung, (messdatenbasierendem) Trace, gemäß Prozedur oder konstant

R-Zustände  $R_i$ : Erzeugung von Aufträgen (genau) eines speziellen Typs  $T(i)$   
 → Werte für Auftragsattribute zu bestimmen, z.B. gemäß Trace, Prozedur oder Verteilung

➤  $\varphi_b$ :

S-Zustände: Blockierungssituation wegen abzuwartender Ereignisse (Systemreaktionen, Betriebsmittelfreigabe, o.ä.)

D-Zustände: Wartezeit für Benutzer beim Warten auf zusätzliche Ereignisse

➤  $\varphi_i, \varphi_t$ : trivial

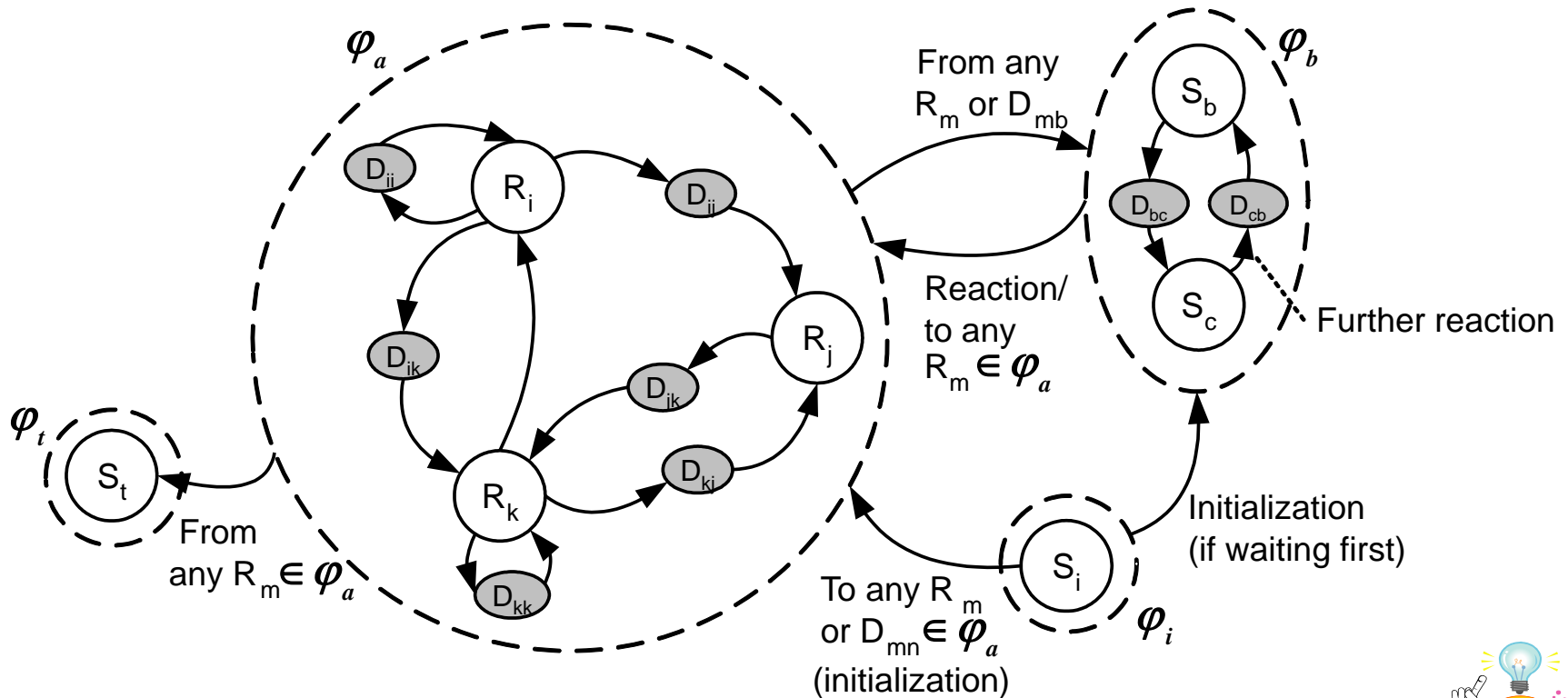
➤  $T_k$ :

Transitionen zwischen Zuständen gesteuert durch:

- Übergangswahrscheinlichkeiten oder Traces
- (optional) abhängig von den Reaktionen des Kommunikationssystems



# Verfeinerung von $\varphi_a$ und $\varphi_b$ durch einen Erweiterten endlichen Automaten $\{R, S, D, T_k\}$



**$R$**  : Menge der **auftragsgenerierenden Zustände**

**$S$**  : Menge der **blockierten Zustände** (spezielle Ereignisse abzuwarten)

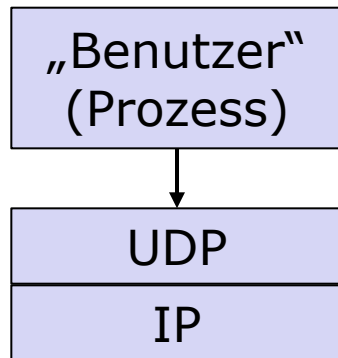
**$D$**  : Menge der Zustände zur **Berücksichtigung benutzerspezifischer Verzögerungen**

**$T_k$**  : Menge der **Transitionen zwischen  $R$ -,  $S$ - und  $D$ -Zuständen** innerhalb eines Makrozust.

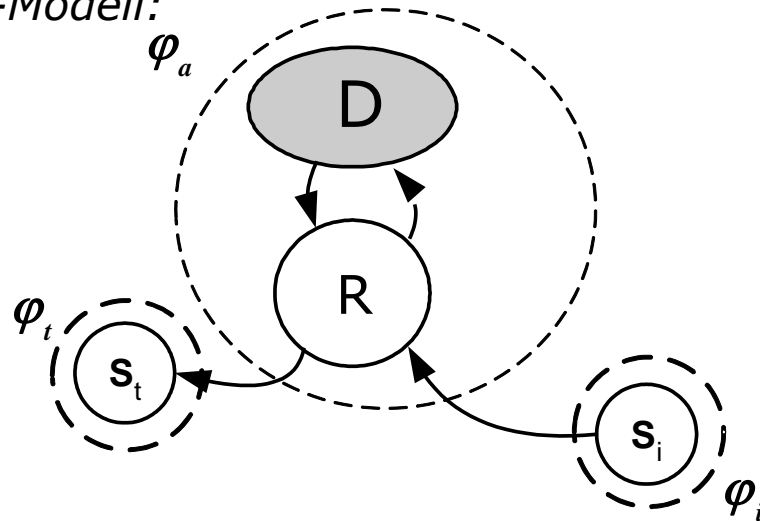
# Beispiel eines sehr einfachen Benutzerverhaltensautomaten:

## BVA eines UDP-“Benutzers“

Protokollstapel:



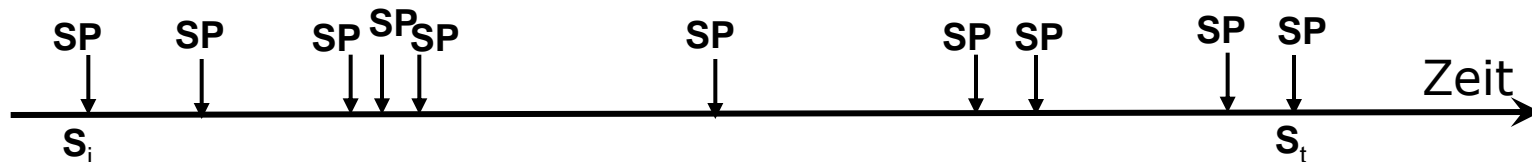
BVA-Modell:



Notation:

- in R generierter Auftrag : SEND\_PACKET (kurz: SP)
- in D modellierte Verzögerung : Verzögerung zwischen 2 aufeinanderfolgenden Sendeaufträgen
- $S_i$  bzw.  $S_t$  : Initialisierungs- bzw. Terminierungszustand

Resultierende Auftragssequenz (Beispiel):



# Lastcharakterisierung

## ➤ Charakterisierung **realer Lasten versus künstliche Lasten**

- **Reallast** interpretierbar und verarbeitbar durch reales Netz  
→ direkte Anpassbarkeit an vorgegebene Systemschnittstelle notwendig bei Wunsch der konkreten Ver-/Bearbeitung von Reallasten

oder

- **Reallast** nur Ausgangspunkt für weitere Abstraktionsvorgänge  
→ z.B. Ausgangspunkt für Gewinnung realitätsnaher Lastmodelle

oder

- **Künstliche Last** evtl. interpretierbar und verarbeitbar durch reales Netz  
→ wiederum Anpassbarkeit an gegebene Systemschnittstelle erforderlich

oder

- **Künstliche Last** evtl. nur relevant für (Simulations-) Modell  
→ Vereinfachungen an modellierter Systemschnittstelle möglich

## Wie BVA-/UBA-Parameterwerte gewinnen ? (BVA: Benutzerverhaltensautomat; UBA: User Behaviour Automaton)

➤ **Charakterisierung der Lastparameter** insbesondere mittels

- **Trace** (Aufzeichnung beobachteter/gemessener Wertesequenzen)

oder

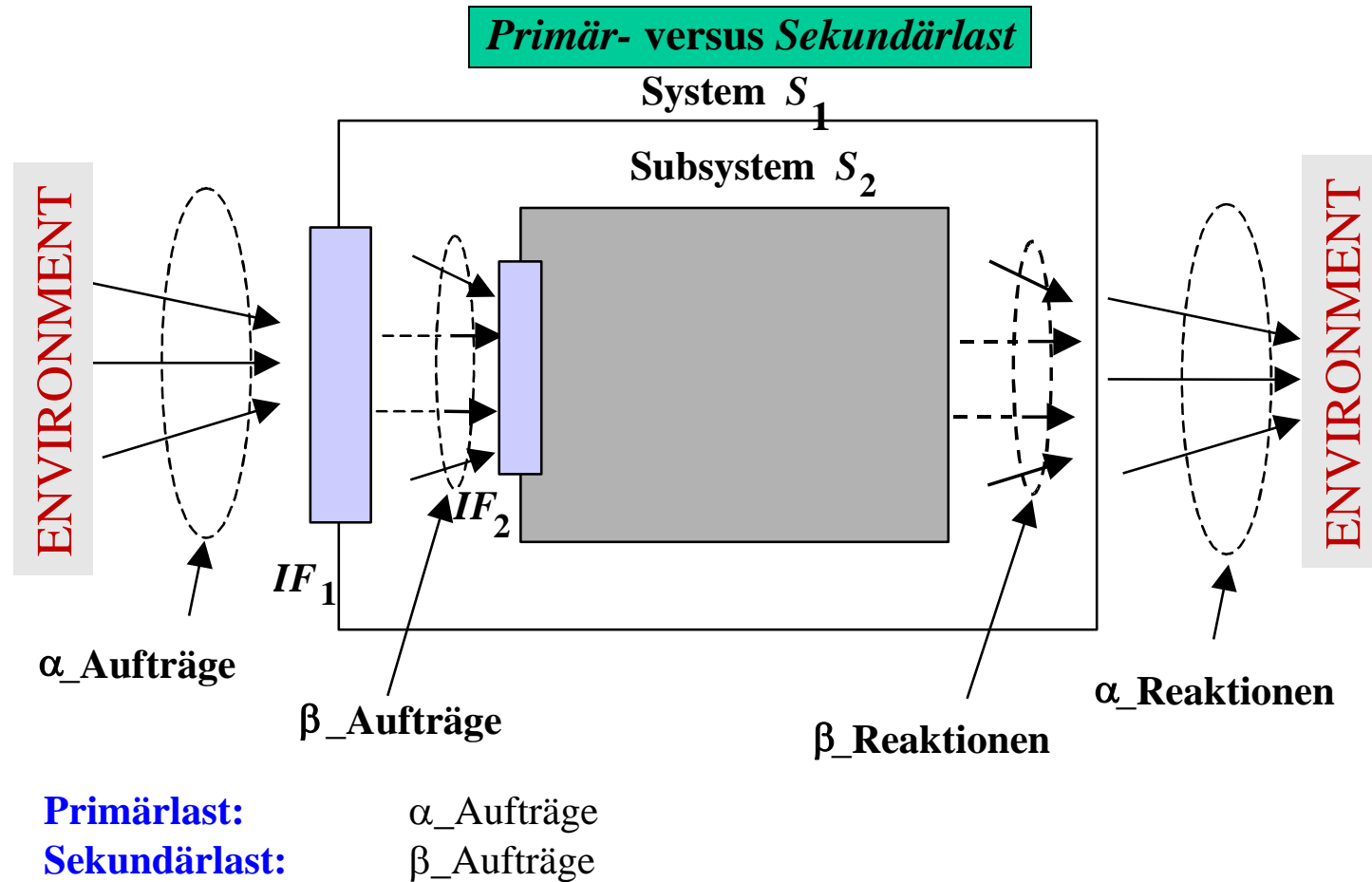
- **Wahrscheinlichkeitsverteilungen** (Zufallszahlgenerierung erforderlich zur Ermittlung konkreter Lastparameterwerte)

⇒ *Lastparameter* betreffen u.a.

- *Auftragsankunftsprozess* (z.B. Zwischenankunftszeiten)
- *Auftragstypen* (z.B. Anzahl aufeinanderfolgender Sendeaufträge)
- *typspezifische Auftragsattribute* (z.B. Längen zu übertragender Dateneinheiten)



# 10.3 Lastmessungen in Rechnernetzen und bei Medienkommunikation



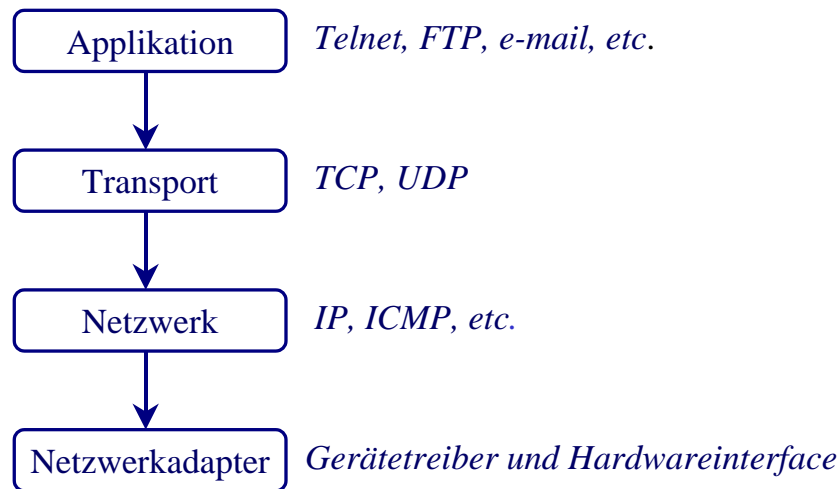
*Wir sagen auch:* Durch das System  $S_1$  wird die Primärlast (an  $IF_1$  beobachtbar) in eine (an  $IF_2$  beobachtbare) Sekundärlast transformiert und nennen diesen Vorgang

**LASTTRANSFORMATION**

**Beispiele :**

- **Primärlast:** zu übertragender digitalisierter und komprimierter *Videostrom* (z.B. MPEG- oder H.261-codiert)
- **Sekundärlast:** zu übertragende *IP-Pakete*
- **Tertiärlast:** zu übertragende *ATM-Zellen* oder *Ethernet-Frames*

Illustration an typischer Protokollhierarchie:



nota bene :

Lastmessungen möglich an unterschiedlichen Schnittstellen  
→ Primär-, Sekundär-, Tertiärlastmessungen

# Grundsätzliche Vorgehensweise bei (Last-) Messungen :



## 1. *Was interessiert ?*

→ insbesondere Motivation/allg. Zielsetzung für Messungen

## 2. *Festlegung der Randbedingungen* für Messungen

→ hier u.a. relevant :

- an welcher **Schnittstelle** gemessen ?
- welche Art von Benutzern vorausgesetzt –  
**reales** versus **künstliches Benutzerverhalten**
- welches **Beobachtungsintervall** ?

nota bene :  $L = L(U, S, IF, T)$

## 3. *welche Messgrößen* erfassen ?

## 4. *Erfassung von Rohdaten*

(reicht zeitliche Auflösung der verwendeten Uhr bei Zeitmessungen ?  
verfälscht der evtl. Betriebsmittelbedarf für die Ausführung von Messprogrammen die Messgrößen ?)

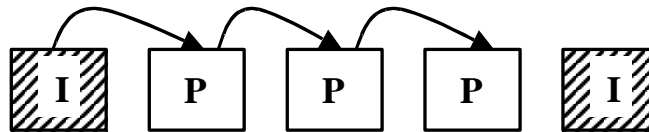
## 5. *Statistische Auswertung* und *graphische Aufbereitung der Rohdaten*

## 6. *Interpretation* und *Weiterverarbeitung* (z.B. Nutzung für eine Lastmodellierung) *der Messresultate*

## (Kleine) Fallstudie zu Lastmessungen :

### Primärlastmessungen bei Videokommunikation

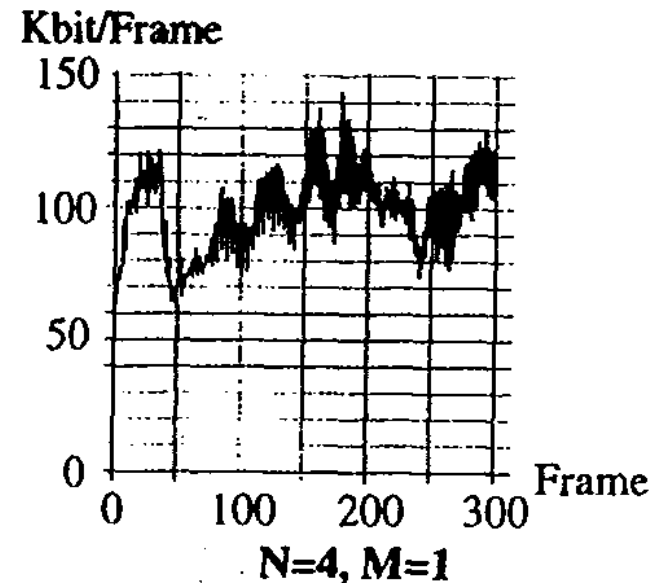
- Betrachteter Videokomprimierungsalgorithmus : MPEG
- Komprimierungsmuster : I P P P I P P P ... ; d.h. N=4 (GOP size) und M=1



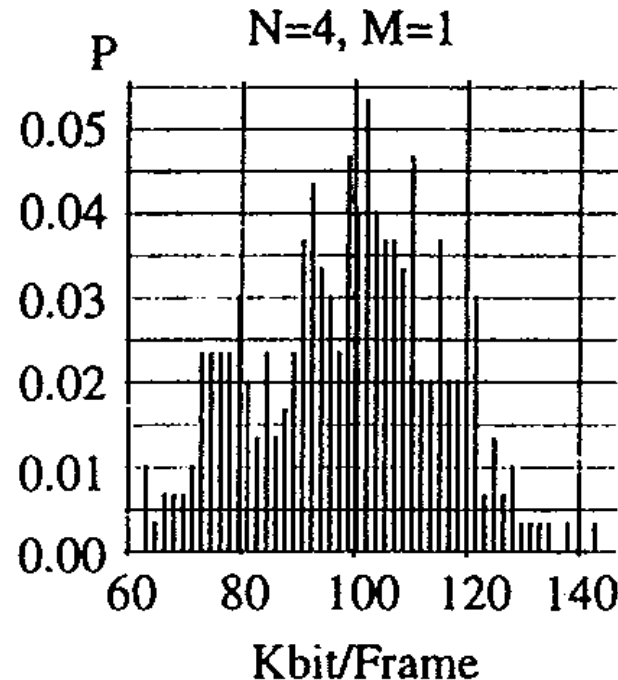
**P** ≡ Predictive Frame; **I** ≡ Intraframe

- Eishockey-Spiel
- Messung ( "**Rohdaten**" ) :

sukzessive auftretende Framelängen



- Aus o.g. Rohdaten **resultierendes Histogramm**  
( $\equiv$  relative Häufigkeit  $p$  von Framelängen bezogen auf äquidistante Framelängenintervalle)



*nota bene :*

- Messungen der Zwischenankunftszeiten der Videoframes evtl. verzichtbar, da i.a. weitestgehend durch Bildwiederholfrequenz determiniert
- Procedure bei Lastmessungen an Sekundärlastschnittstelle (z.B. zur Ermittlung der Länge von IP-Paketen) ganz entsprechend

# 10.4 Lastmodellierung für Rechnernetzbenutzer

## Modellierungsanforderungen

**Allgemeine Anforderungen** bei der Lastmodellierung:

### Charakterisiere

- **Ankunftsprozess** der Aufträge (an  $IF$  während  $T$ )
- **Betriebsmittelbedarfe** und **andere relevante Auftragsattribute**

*nota bene :*

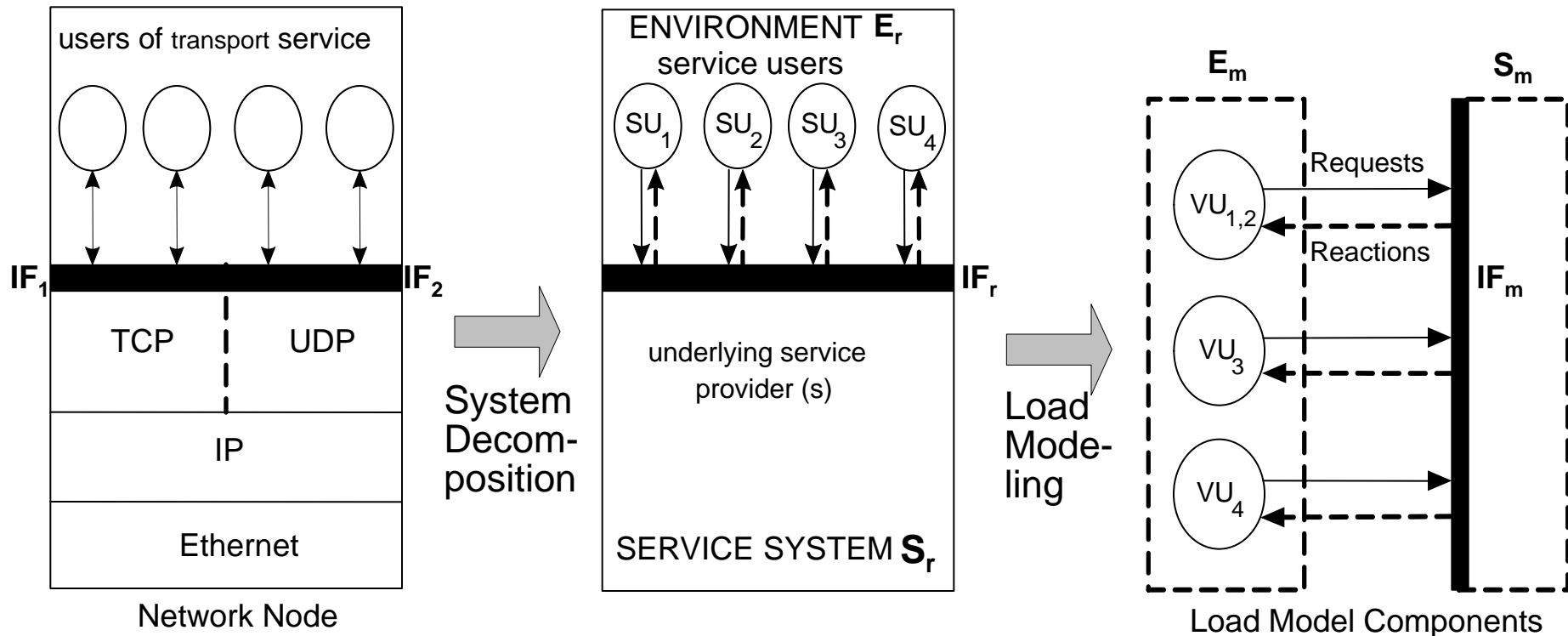
Lastmodelle erforderlich bei Leistungsbewertungen sowohl mittels simulativer als auch math.-analytischer Modelle.

**Spezielle Anforderungen** an ein generalisiertes Procedere zur Lastmodellierung (insbesondere zur Gewinnung von Lastmodellen für Simulationsexperimente) :

- *Unterstützung der (schrittweisen) Verfeinerung* eines gegebenen Lastmodells:
  - zusätzliche Auftragsstypen
  - zusätzliche Attribute für Typen von Aufträgen
  - zunehmend präzise Berücksichtigung von Attributwerten.
- *Unterstützung der Vergrößerung* eines gegebenen Lastmodells.
- Gestatte die Erarbeitung von *Lastmodellen für verschiedene Schnittstellen* (unter Nutzung derselben vereinheitlichten Modellierungsmethode).
- Ermögliche *hinreichend repräsentative Lastmodelle* (z.B. unter Rückgriff auf Lastmessungen).
- *Reflektiere* mögliche *Abhängigkeiten von U von dem Momentanzustand von S* während der Lastgenerierung (z.B. *U* temporär blockiert).

# Veranschaulichung unserer Definition von Last sowie des Procederes bei Lastmodellierung

*Last*





# Verallgemeinertes Procedere zur Lastmodellierung



## SCHRITT 1: Dekomposition von System und Umgebung

(sowohl für das reale Netz als auch im Modellbereich)

- Zerlege  $S$  und  $U$  ( $\rightarrow$  wohldefinierte Schnittstelle  $IF$ , schwache Interaktionen).
- Entscheide, welche Aufträge zu berücksichtigen sind (an  $S$  über  $IF$  übergeben).
- Bilde  $U$  ab auf Menge lastgenerierender “Benutzer”  $B_1, B_2, \dots, B_n$  (d.h. last- bzw. auftragsgenerierende Quellen).

## SCHRITT 2: Wahl eines adäquaten Detaillierungsgrades der Modellierung

$\Rightarrow$  Verwendung eines *objektorientierten Ansatzes* zur Modellierung von Aufträgen

- Bilde disjunkte Klassen von Aufträgen  
(Klasse  $\equiv$  eindeutiger *Auftragstyp* ) .
- Lege typspezifische Menge von *Auftragsattributen* fest.

## Verallgemeinertes Procedere zur Lastmodellierung (Forts.)

### SCHRITT 3: Analyse und Beschreibung der grundsätzlich möglichen Interaktionen (für gegebenes $IF$ )

⇒ Spezifiziere mögliche Sequenzen von Interaktionen zwischen  $S$  und  $U$  (vgl. Dienstspezifikation für Kommunikationsdienste).

### SCHRITT 4: Beschreibung der tatsächlichen Interaktionen zwischen $S$ und $U$ (während des Beobachtungsintervalls $T$ )

- Bilde jeden Benutzer  $B_i$  (vgl. SCHRITT 1) auf einen **individuellen Lastgenerator** ab, der einen Auftragsstrom  $S_R$  erzeugt, wobei

$$S_R = ((t_1, R_1), (t_2, R_2), (t_3, R_3), \dots, (t_m, R_m)), \quad t_i \leq t_{i+1},$$

wobei  $t_i$  den Ankunftszeitpunkt von Auftrag  $R_i$  an  $IF$  bezeichne.

- Spezifiziere die Folge  $(t_1, t_2, t_3, \dots, t_m)$  mittels
  - Trace (*deterministischer* Ankunftsprozess), oder
  - Wahrscheinlichkeitsverteilung (*stochastischer* Ankunftsprozess).



**5 % Paketverluste,  
keine Fehlertoleranz**

**Bild-Nr.: 341  
119**



**5 % Paketverluste, Nutzung lastabhängiger Videocodierung zur Fehlertoleranz**



# Lastmessungen und Längenapproximationen durch Normalverteilung

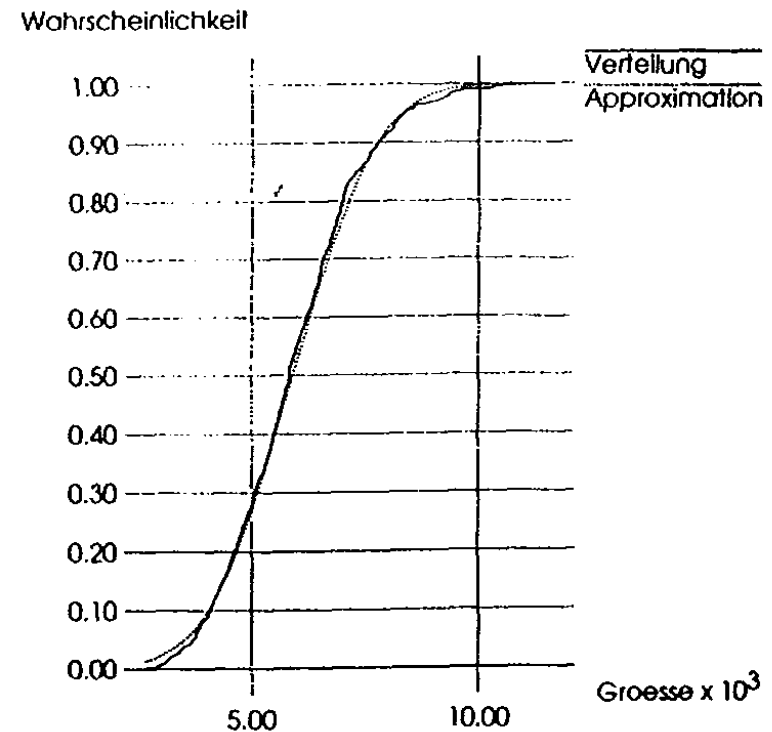
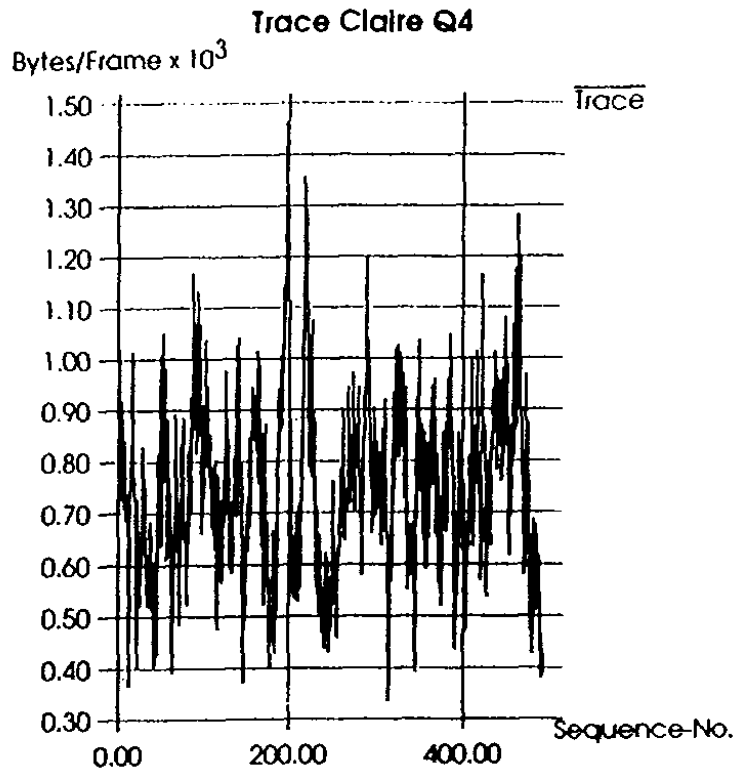


Abbildung : H.261-Trace Claire Q4 (links) und Framelängenverteilung (rechts)  
[Q4 für gewählten Quantisierungsfaktor]



Zur Beurteilung der **Güte von Approximationen** :

→ **math. Testverfahren** der Statistik anzuwenden

(bei Approximationen durch Normalverteilung, s.o.,  
z.B. **Kolmogorov**- oder  $\chi^2$ - Test)

## Resümee :

**Mögliche Schritte bei Lastmodellierung** am Beispiel einer Modellierung des Auftrag-sattributes "Länge" (von Videoframes)

1. **Aufzeichnung** sukzessiver **Framelängen** (bei hinreichend langer Videosequenz)
2. **Ermittlung** der resultierenden **Häufigkeitsverteilung** (Histogramm)
3. **Wahl** einer, für eine Approximation geeignet erscheinenden, **Klasse von Wahrscheinlichkeitsverteilungen**
4. **"Optimale" Parametrisierung** der gewählten Klasse von Wahrscheinlichkeitsverteilungen  
(z.B.  $\mu$  und  $\sigma$  freie Parameter)  
→ dieser Schritt entspricht der Verteilungs-Approximation
5. **Bewertung** der erzielten **Güte der Approximation** mittels Testverfahren.



# Das eLearning-Werkzeug **LoadSpec**

## zur realitätsnahen Lastspezifikation für Rechnernetzbenutzer



→ **LoadSpec** (*“Load Specifier”*) gestattet :

- die Spezifikation von Benutzerverhaltensautomaten (**BVAs**), unter Nutzung der formalen Lastbeschreibungstechnik aus Abschn. 10.2
- die konkrete Parametrisierung der spezifizierten Automaten (→ **parametrisierter BVA – pBVA**), insbesondere bzgl. Auftragsankunftsprozess sowie Auftragsattributwerten
- die Generierung und Veranschaulichung von Auftragssequenzen (u.a. Auftragsankunftsstrom sowie dynamischer Verlauf wesentlicher Auftragsattribute, wie z.B. Paketlängen)

Erfahrungen mit der konkreten Nutzung / Anwendung von **LoadSpec** in den DKR-Übungen/Praktikum ... und Präsentation des Tools in der DKR-Vorlesung !

*Nota bene:* Lastspezifikation gemäß *LoadSpec* kann als Basis für eine Lastgenerierung an unterschiedlichen Schnittstellen eines Netzes dienen

→ vgl. hierzu das Werkzeug **UniLoG – Unified Load Generator** (ein verallgemeinerter Lastgenerator), siehe u.a. F&E von *Andrey Kolesnikov (TKRN-Doktorand)* und *Prof. Dr. Jing Cong* und *viele Studierende unseres FBIs (bislang ≈ 10)*

# eLearning-Werkzeug LoadSpec :

## Spezifikation eines *Voice-over-IP*-Modells (incl. Parametris.)

LSPEC: Lastspezifikation gemäß TKN-Ansatz - TalkSpurt.pbva \*

Datei Bearbeiten Experimentreihe Ansicht Fenster Hilfe

**TalkSpurt.pbva**

- BVA: D:\Informatik\Studienarbeit\Deploym
  - INITIALIZE (MACRO)
  - I1
  - ACTIVE (MACRO)
    - RC
      - Start\_Call
        - AT
          - DestIP
          - DestPort
          - DataLength

**TalkSpurt.pbva \***

- PBVA: D:\Informatik\Studienar
  - INITIALIZE (MACRO)
  - I1
  - ACTIVE (MACRO)
    - RC
      - Start\_Call
        - AT
          - DestIP
          - DestPort
          - DataLength
      - DC
        - RT
          - Send\_Packet
            - AT
              - DestIP
              - DestPort
              - DataLength
              - n
              - f
              - 0x
              - AA
              - bitpattern
          - DH
            - RH
              - DH
                - Hang\_Up
                  - AT
                    - DestIP
                    - DestPort
                    - Datenlänge

**Eingabe der Werte der Auftragsattribute**

Attribute des zugeordneten Auftragstyps:

    - RT
      - Send\_Packet
        - AT
          - ip DestIP
          - DestPort
          - DataLength
          - n
          - f
          - 0x
          - AA
          - bitpattern

Attributwert gemäß

    - ☐ Konstant
    - ☒ Verteilung
    - ☐ Trace

Gleichverteilt in (a,b)  
Exponentialverteilt (Lambda)  
Erlang-verteilt (Lambda, k)  
Normalverteilt (m, s)

Spaltennummer: 1

OK Abbrechen

```

graph LR
    I1((I1)) -- 1.000000 --> RC((RC))
    RC -- 0.100000 --> DR((DR))
    DR -- 1.000000 --> RC
    RC -- 0.899800 --> DW((DW))
    DW -- 1.000000 --> RC
    RC -- 0.000200 --> DC((DC))
    DC -- 1.000000 --> RC
    RC -- 1.000000 --> RH((RH))
    RH -- 1.000000 --> RC
    RH -- 1.000000 --> DH((DH))
    DH -- 1.000000 --> RT((RT))
    RT -- 0.000020 --> DH
    RT -- 0.997980 --> DS((DS))
    DS -- 1.000000 --> RT
    RT -- 0.002000 --> DT((DT))
    DT -- 1.000000 --> RT
  
```

Bereit

Start D:\Informatik\Stu... LSPEC: Lastspez... Unbenannt - Paint DE 16:29

# eLearning-Werkzeug LoadSpec :

## *MPEG-Modell* und daraus resultierende Framelängen über der Zeit

