Peer-to-Peer-Netzwerke

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer

Sommersemester 2006

4. Vorlesung

04.05.2006

schindel@informatik.uni-freiburg.de

Inhalte

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Kurze Geschichte der Peer-to-Peer-Netzwerke**
- **Das Internet: Unter dem Overlay**
- **Die ersten Peer-to-Peer-Netzwerke**
  - Napster
  - Gnutella
  - Die Verbindungsstruktur von Gnutella
- **Chord**
- **Pastry und Tapestry**
- **Gradoptimierte Netzwerke**
  - Viceroy
  - Distance-Halving
  - Koorde
- **Netzwerke mit Suchbäumen**
  - Skipnet und Skip-Graphs
  - P-Grid
- **Selbstorganisation**
  - Pareto-Netzwerke
  - Zufallsnetzwerke
  - Selbstorganisation
  - Metrikbasierte Netzwerke Sicherheit in Peer-to-Peer-Netzwerken
- **Anonymität**
- **Datenzugriff: Der schnellere Download**
- **Peer-to-Peer-Netzwerke in der Praxis**
  - eDonkey
  - FastTrack
  - Bittorrent
- **Peer-to-Peer-Verkehr**
- **Juristische Situation**

Die Internet-Schichten TCP/IP-Layer

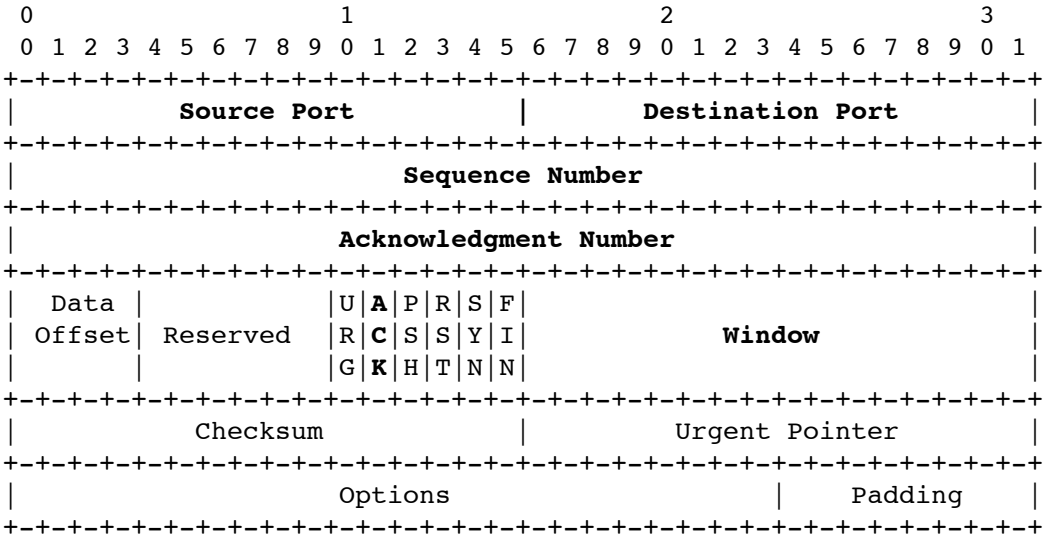
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Anwendung	Application	Peer-to-Peer-Netzwerke, HTTP (Web), SMTP (E-Mail), ...
Transport	Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Vermittlung	Network	IP **(Internet Protocol)** + ICMP **(Internet Control Message Protocol)** + IGMP **(Internet Group Management Protoccol)**
Verbindung	Link	LAN **(z.B. Ethernet, Token Ring etc.)**

TCP-Header

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

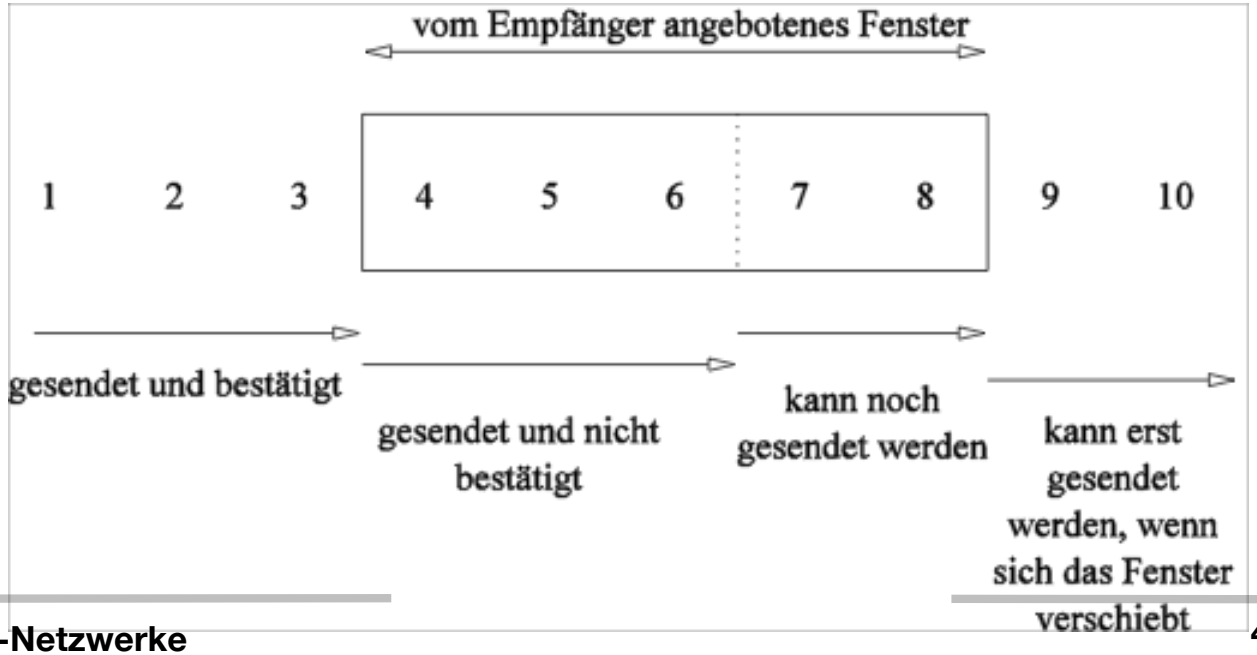
- **Sequenznummer**
  - Nummer des ersten Bytes im Segment
  - Jedes Datenbyte ist nummeriert modulo 256
- **Bestätigungsnummer**
  - Aktiviert durch ACK-Flag
  - Nummer des nächsten noch nicht bearbeiteten Datenbytes
    - = letzte Sequenznummer + letzte Datenmenge
- **Sonstiges:**
  - Port-Adressen
    - Für parallele TCP-Verbindungen
    - Ziel-Port-Nr.
    - Absender-Port
  - Headerlänge
    - data offset
  - Prüfsumme
    - Für Header und Daten



Gleitende Fenster (sliding windows)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

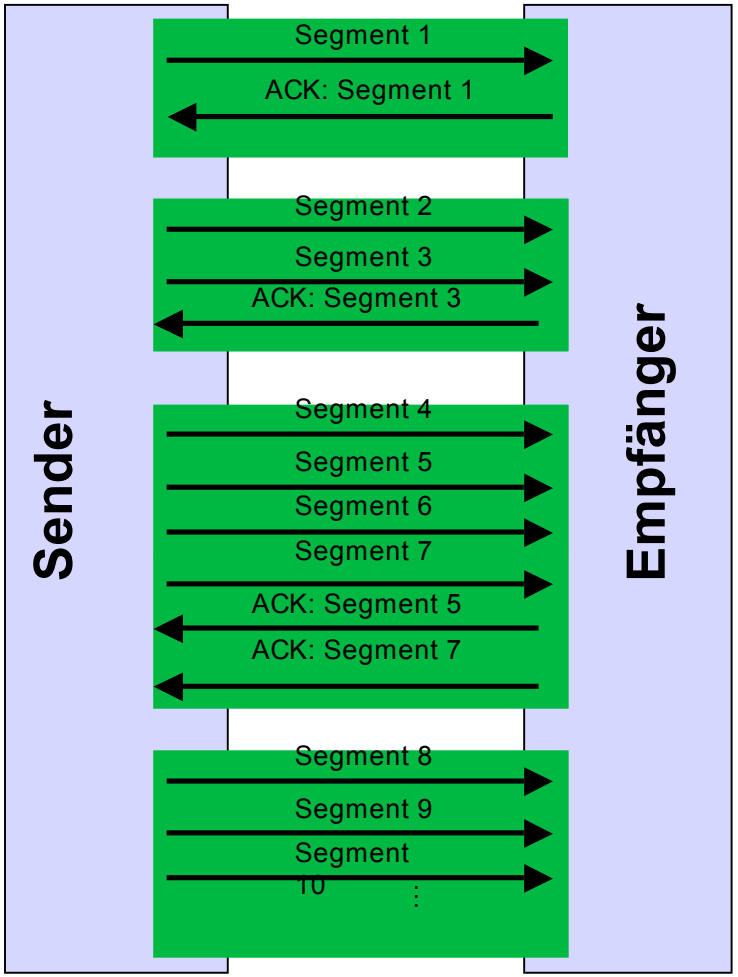
- **Datenratenanpassung durch Fenster**
  - Empfänger bestimmt Fenstergröße (wnd) im TCP-Header der ACK-Segmente
  - Ist Empfangspuffer des Empfängers voll, sendet er wnd=0
  - Andernfalls sendet Empfänger wnd>0
- **Sender beachtet:**
  - Anzahl unbestätigter gesender Daten ≤ Fenstergröße



Slow Start
Congestion Fenster

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Sender darf vom Empfänger angebotene Fenstergröße nicht von Anfang wahrnehmen**
- **2. Fenster: Congestion-Fenster (cwnd/Congestion window)**
  - Von Sender gewählt (FSK)
  - Sendefenster: min {wnd,cwnd}
  - S: Segmentgröße
  - Am Anfang:
    - cwnd ← S
  - Für jede empfangene Bestätigung:
    - cwnd ← cwnd + S
  - Solange bis einmal Bestätigung ausbleibt
- **„Slow Start“ = Exponentielles Wachstum**



Durchsatzoptimierung in der Transportschicht

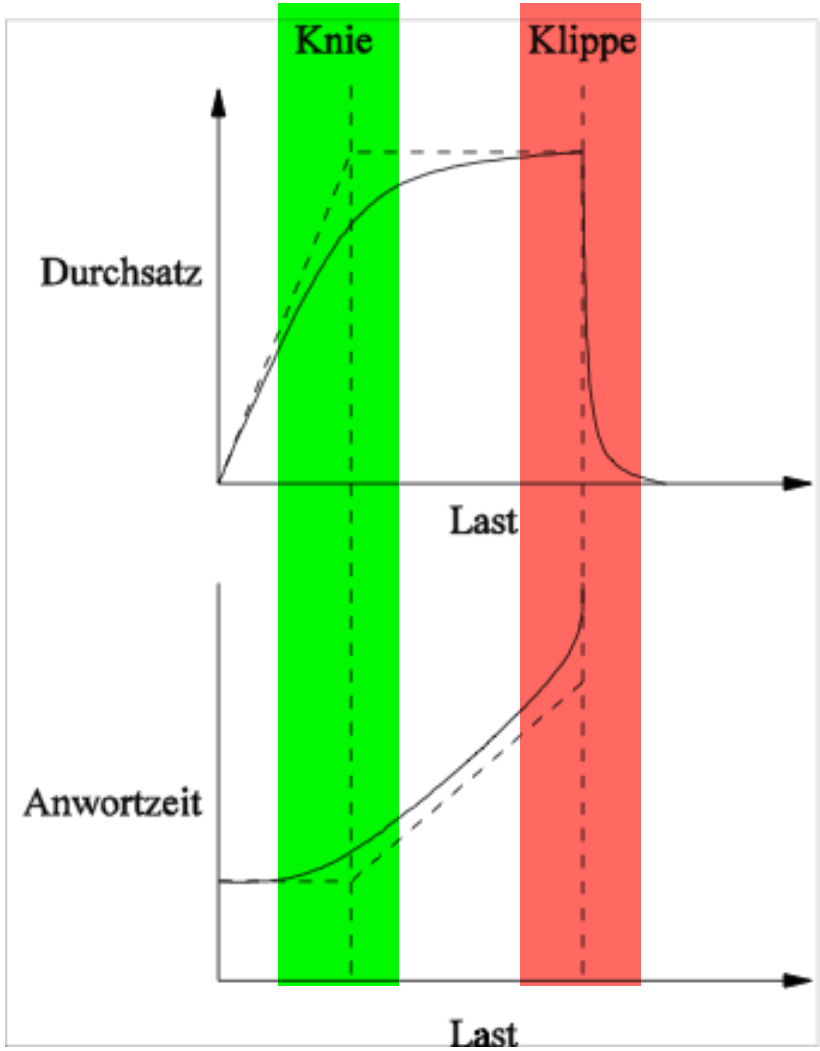
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Hauptproblem von TCP: Durchsatzoptimierung durch Stauvermeidung**
  - Jacobson: min. 99% aller verlorenen Pakete durch IP wegen Datenüberlaufs (congestion) an Routern
  - Feedback in der Transportschicht nur durch Ackn.-Pakete:
    - Wird ein Paket nicht bestätigt, war der gewählte Datendurchsatz zu hoch
      - TCP verringert Datendurchsatz
    - Werden alle Pakete bestätigt, war der Datendurchsatz genau richtig oder zu niedrig
      - TCP erhöht Datendurchsatz

Durchsatz und Antwortzeit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Klippe:**
  - Hohe Last
  - Geringer Durchsatz
  - Praktisch alle Daten gehen verloren
- **Knie:**
  - Hohe Last
  - Hoher Durchsatz
  - Einzelne Daten gehen verloren



Modell

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Rundenmodell**
  - In Runde t steht Bandweite u_t zur Verfügung
  - Algorithmus fordert in Runde Bandweite xt an
  - Feedback zu Beginn von Runde t+1:
    - Ist $x_t > u_t$? (d.h. sind Pakete verloren gegangen)
  - Algorithmus erfährt nie die wirklich verfügbaren Bandweiten

Runde	Host A	Internet	Host B
t	sendet x_t Pakete	Bandweite läßt maximal u_t Pakete durch →	erhält $\min(x_t,u_t)$ Pakete
	erhält Ack. und berechnet x_{t+1}	←	schickt Bestätigung (Ack.)
t+1	sendet x_{t+1} Pakete	Bandweite läßt maximal u_{t+1} Pakete durch →	erhält $\min(x_{t+1},u_{t+1})$ Pakete

Stauvermeidung in TCP Tahoe

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

x: Anzahl Pakete pro RTT

- **Jacobson 88:**
  - Parameter: cwnd und Slow-Start-Schwellwert (ssthresh=slow start threshold)
  - S = Datensegmentgröße = maximale Segmentgröße
- **Verbindungsaufbau:**
  - cwnd ← S ssthresh ← 65535

x ← 1

y ← max

- **Bei Paketverlust, d.h. Bestätigungsdauer > RTO,**
  - Dann *multiplicatively decreasing*
    cwnd ← S ssthresh ← $\max\left\{2S, \frac{1}{2}\min\{\text{cwnd}, \text{wnd}\}\right\}$

x ← 1

y ← x/2

- **Werden Segmente bestätigt und cwnd ≤ ssthresh,**
  - dann *slow start*
    cwnd ← cwnd + S

x ← 2·x, bis x = y

- **Werden Segmente bestätigt und cwnd > ssthresh,**
  - dann *additively increasing*

$$\text{cwnd} \leftarrow \text{cwnd} + \frac{S}{\text{cwnd}}$$

x ← x +1

TCP Tahoe

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

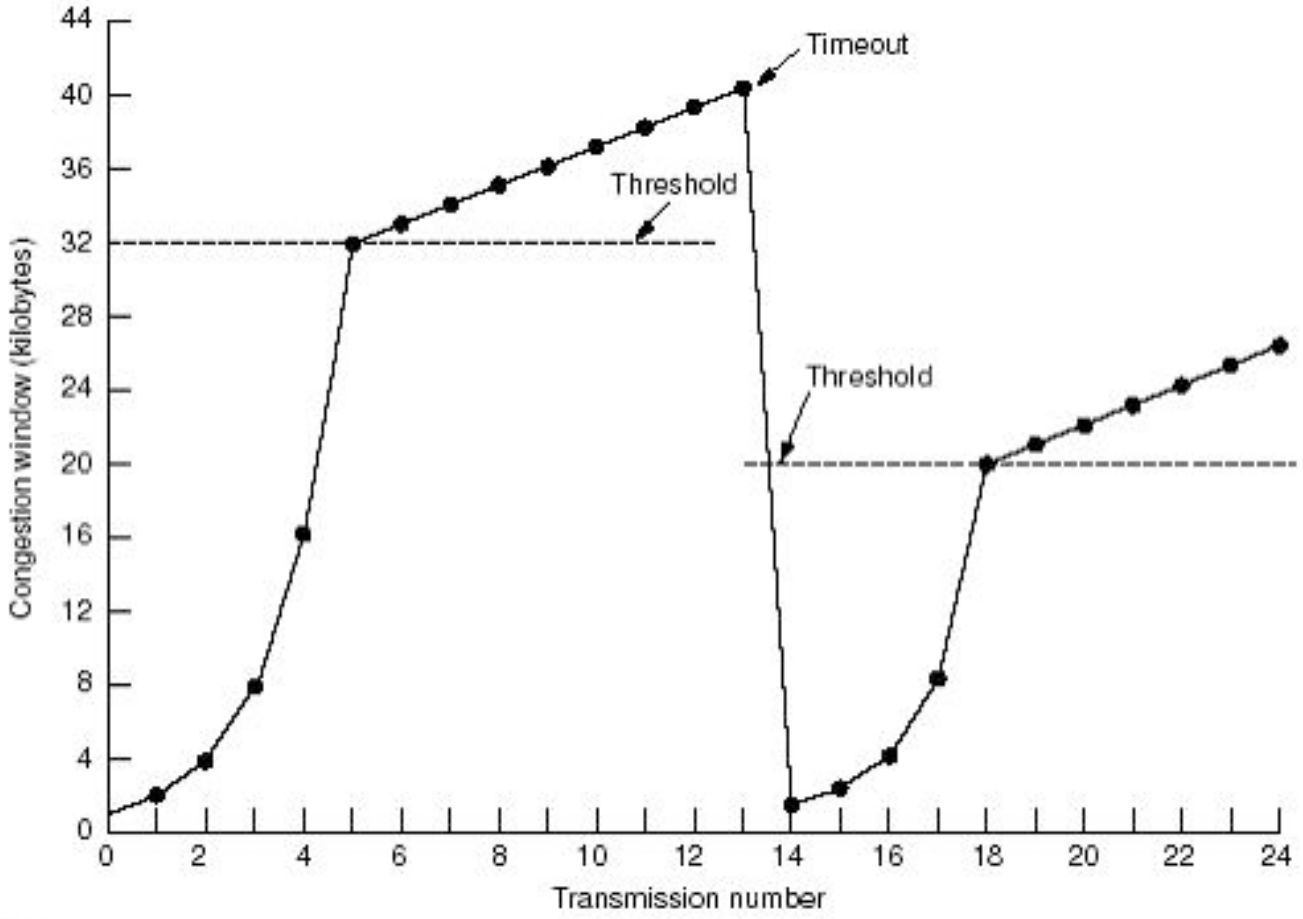


Fig3

pictures from TANENBAUM A. S. *Computer Networks 3rd edition*

Fast Retransmit und Fast Recovery

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **TCP Tahoe [Jacobson 1988]:**
  - Geht nur ein Paket verloren, dann
    - Wiederversand Paket + Restfenster
    - Und gleichzeitig Slow Start
  - Fast retransmit
    - Nach drei Bestätigungen desselben Pakets (triple duplicate ACK),
    - sende Paket nochmal, starte mit Slow Start
- **TCP Reno [Stevens 1994]**
  - Nach Fast retransmit:
    - ssthresh ← min(wnd,cwnd)/2
    - cwnd ← ssthresh + 3 S
  - Fast recovery nach Fast retransmit
    - Erhöhe Paketrate mit jeder weiteren Bestätigung
    - cwnd ← cwnd + S
  - Congestion avoidance: Trifft Bestätigung von P+x ein:
    - cwnd ← ssthresh

y ← x/2

x ← y + 3

Stauvermeidungsprinzip: AIMD

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Kombination von TCP und Fast Recovery verhält sich im wesentlichen wie folgt:**

- **Verbindungsaufbau:**

$$x \leftarrow 1$$

- **Bei Paketverlust, MD:multiplicative decreasing**

$$x \leftarrow x/2$$

- **Werden Segmente bestätigt, AD: additive increasing**

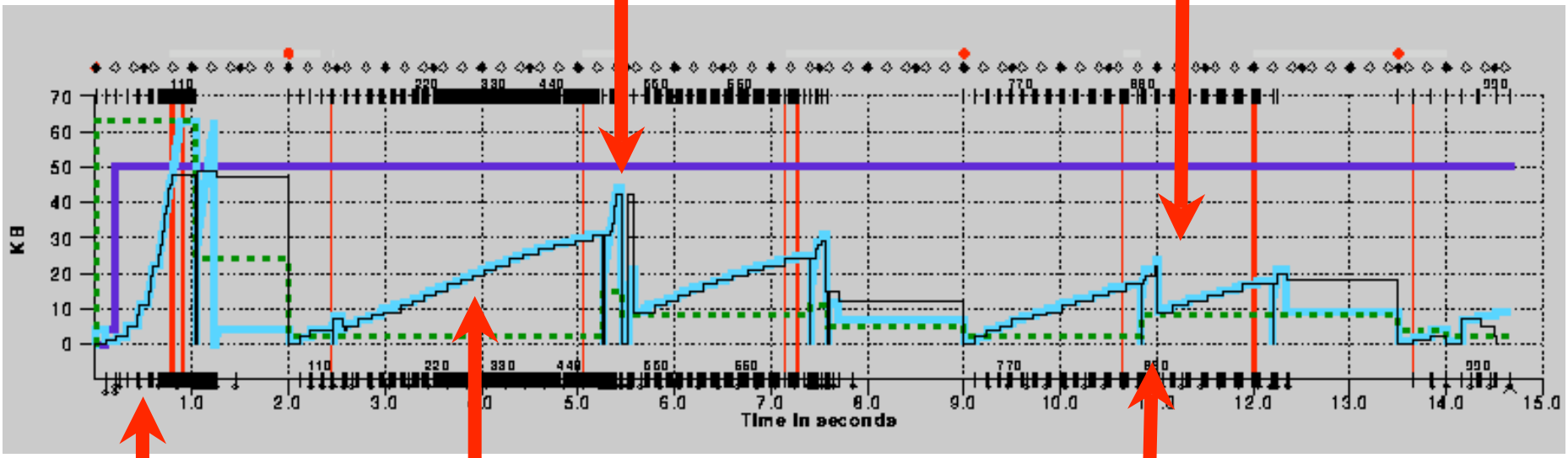
$$x \leftarrow x + 1$$

Beispiel: TCP Reno in Aktion

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Fast Retransmit

Fast Recovery



Additively Increase

Slow Start

Multiplicatively Decrease

Ein einfaches Datenratenmodell

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **n Teilnehmer, Rundenmodell**
  - Teilnehmer i hat Datenrate xi(t)
  - Anfangsdatenrate $x_1(0), \ldots, x_n(0)$ gegeben
- **Feedback nach Runde t:**
  - y(t) = 0, falls $\sum_{i=1}^{n} x_i(t) \leq K$
  - y(t) = 1, falls $\sum_{i=1}^{n} x_i(t) > K$
  - wobei K ist Knielast
- **Jeder Teilnehmer aktualisiert in Runde t+1:**
  - $x_i(t+1) = f(x_i(t),y(t))$
  - Increase-Strategie $f_0(x) = f(x,0)$
  - Decrease-Strategie $f_1(x) = f(x,1)$
- **Wir betrachten lineare Funktionen:**

$$f_0(x) = a_I + b_I x \qquad \text{und} \qquad f_1(x) = a_D + b_D x \ .$$

Lineare Datenratenanpassung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Interessante Spezialfälle:**
  - AIAD: Additive Increase Additive Decrease

$$f_0(x) = a_I + x \quad \text{und} \quad f_1(x) = a_D + x \ ,$$

wobei $a_I > 0$ und $a_D < 0$.

  - MIMD: Multiplicative Increase/Multiplicative Decrease

$$f_0(x) = b_I x \quad \text{und} \quad f_1(x) = b_D x \ ,$$

wobei $b_I > 1$ und $b_D < 1$.

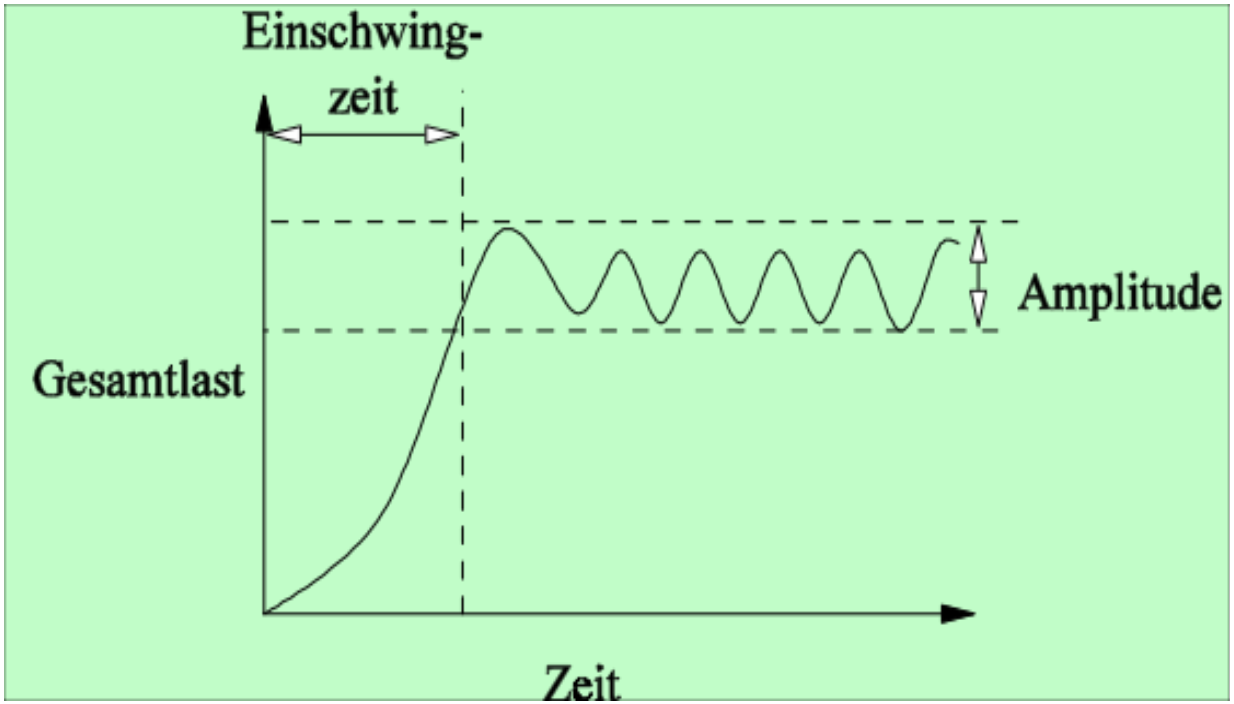
  - AIMD: Additive Increase Multiplicative Decrease

$$f_0(x) = a_I + x \quad \text{und} \quad f_1(x) = b_D x \ ,$$

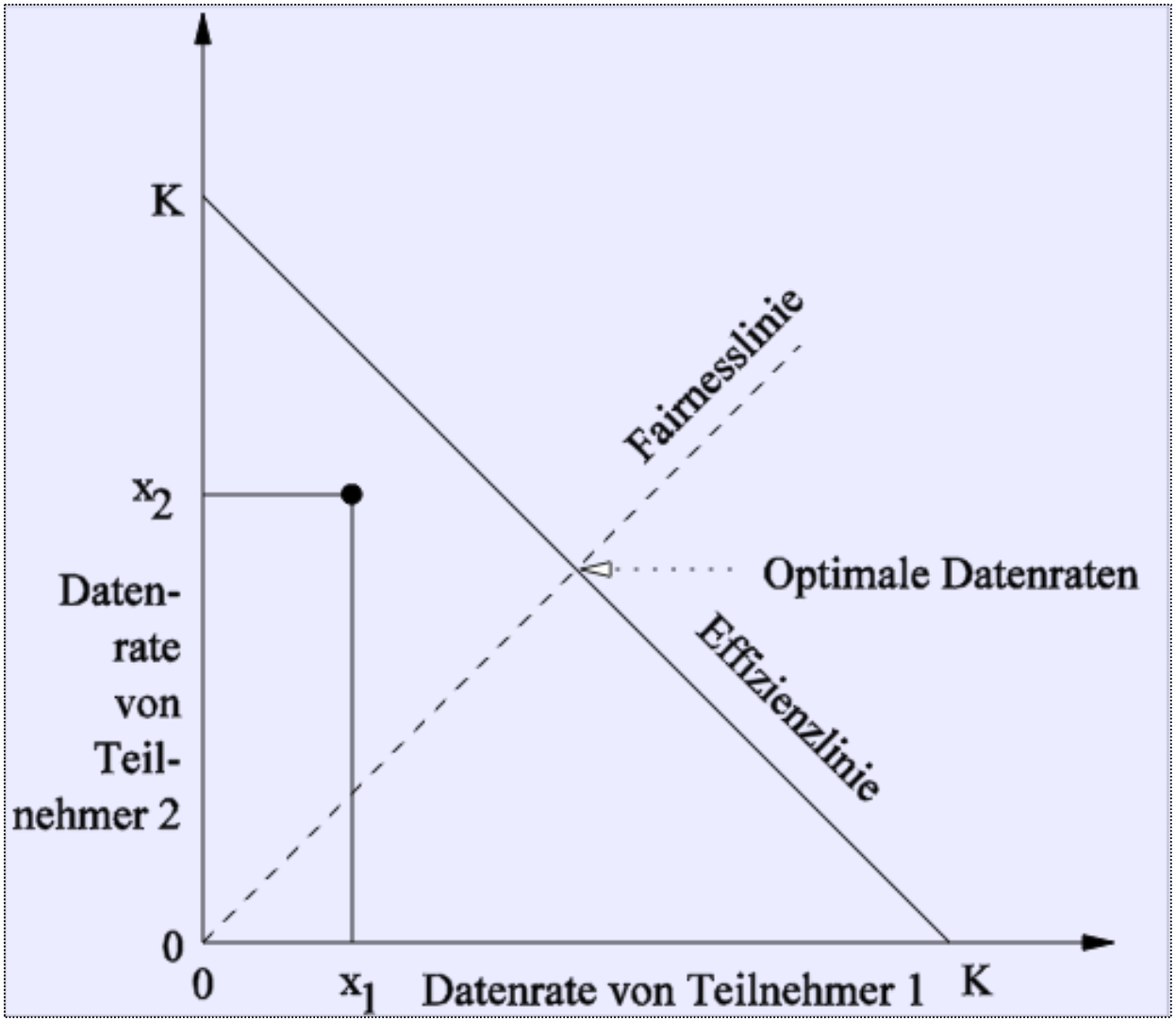
wobei $a_I > 0$ und $b_D < 1$.

Konvergenz

- **Konvergenz unmöglich**
- **Bestenfalls Oszillation um Optimalwert**
  - Oszillationsamplitude A
  - Einschwingzeit T

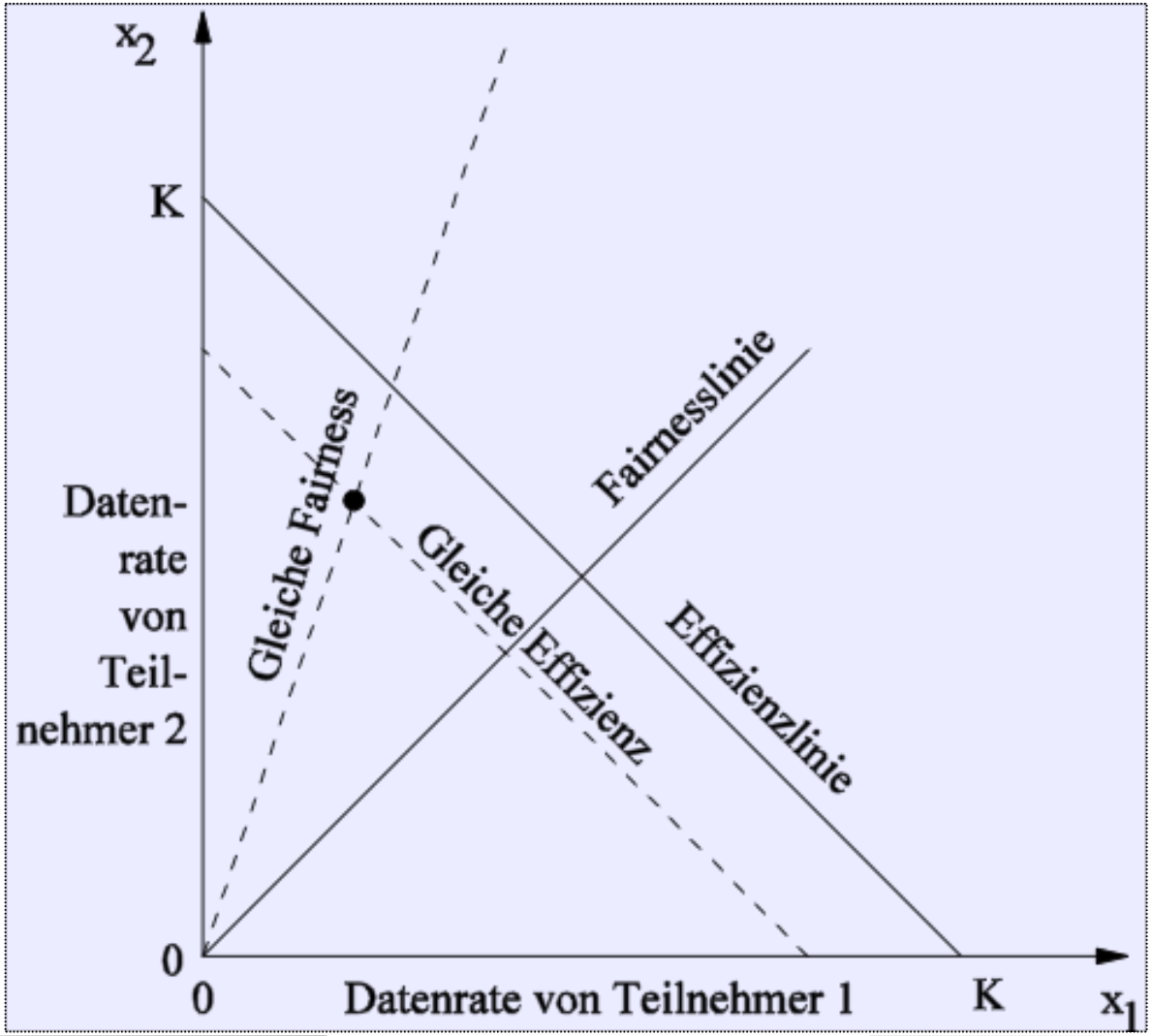


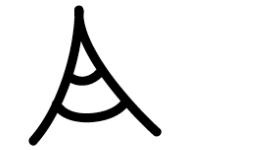
Vektordarstellung (I)



Vektordarstellung (II)

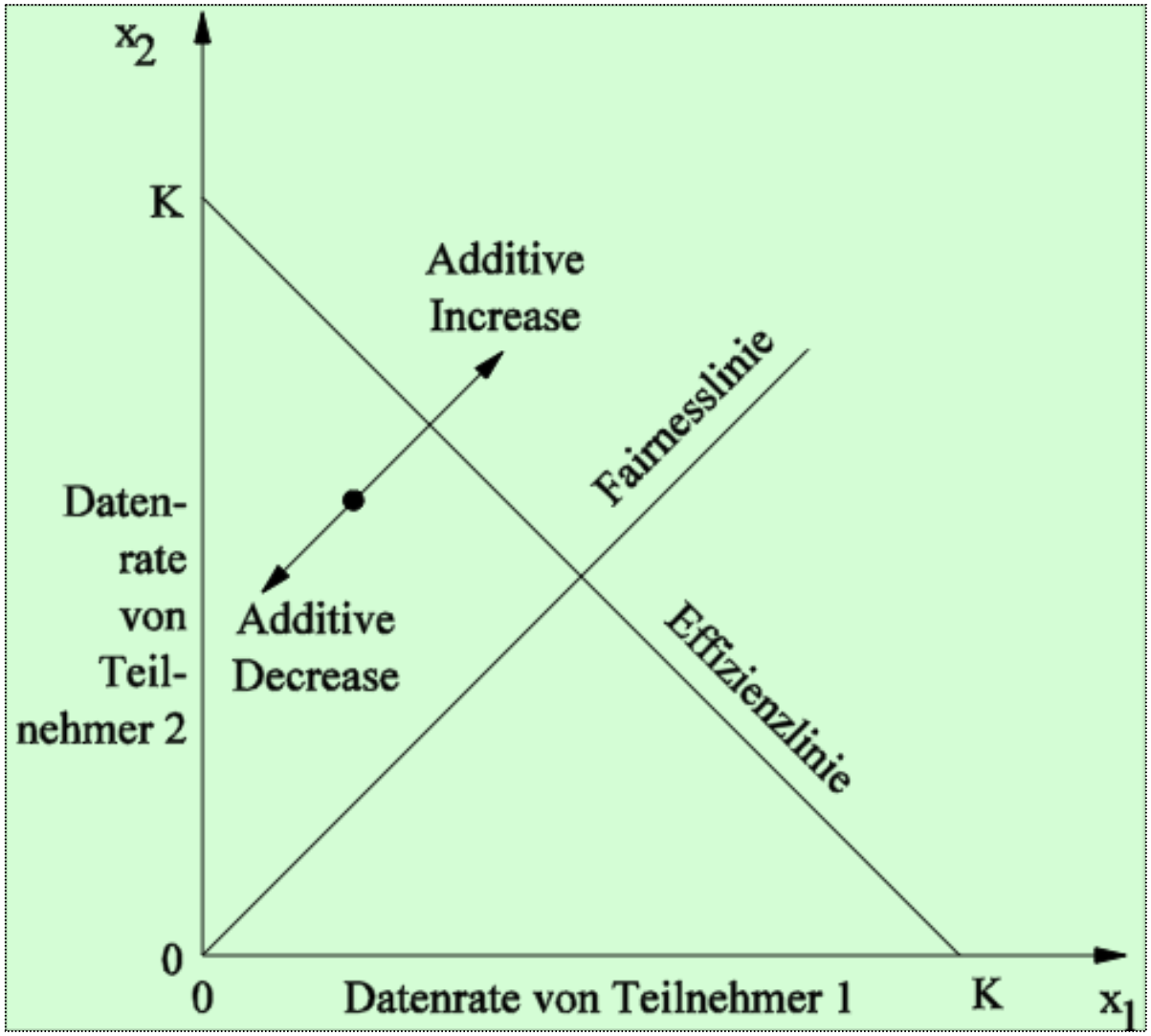
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

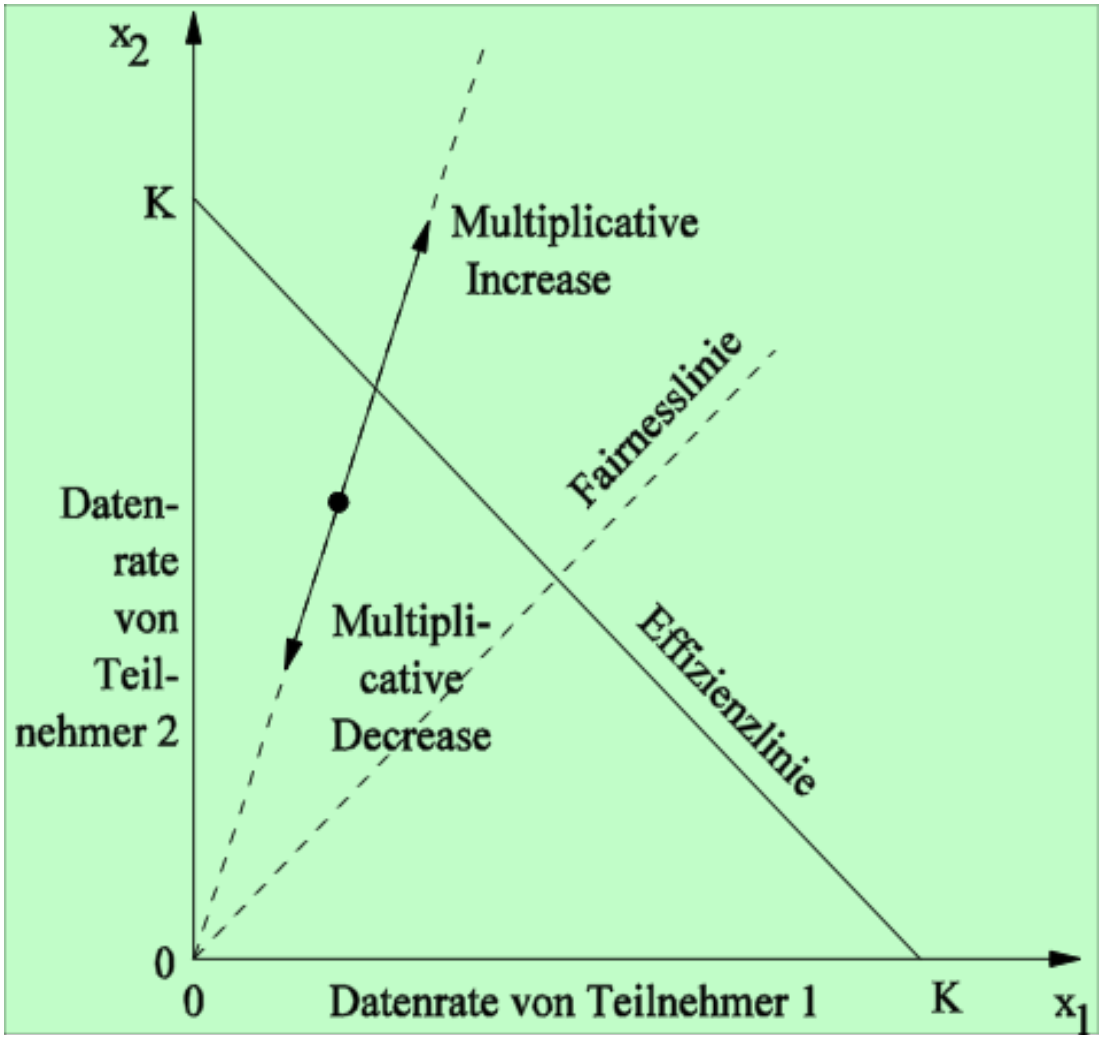


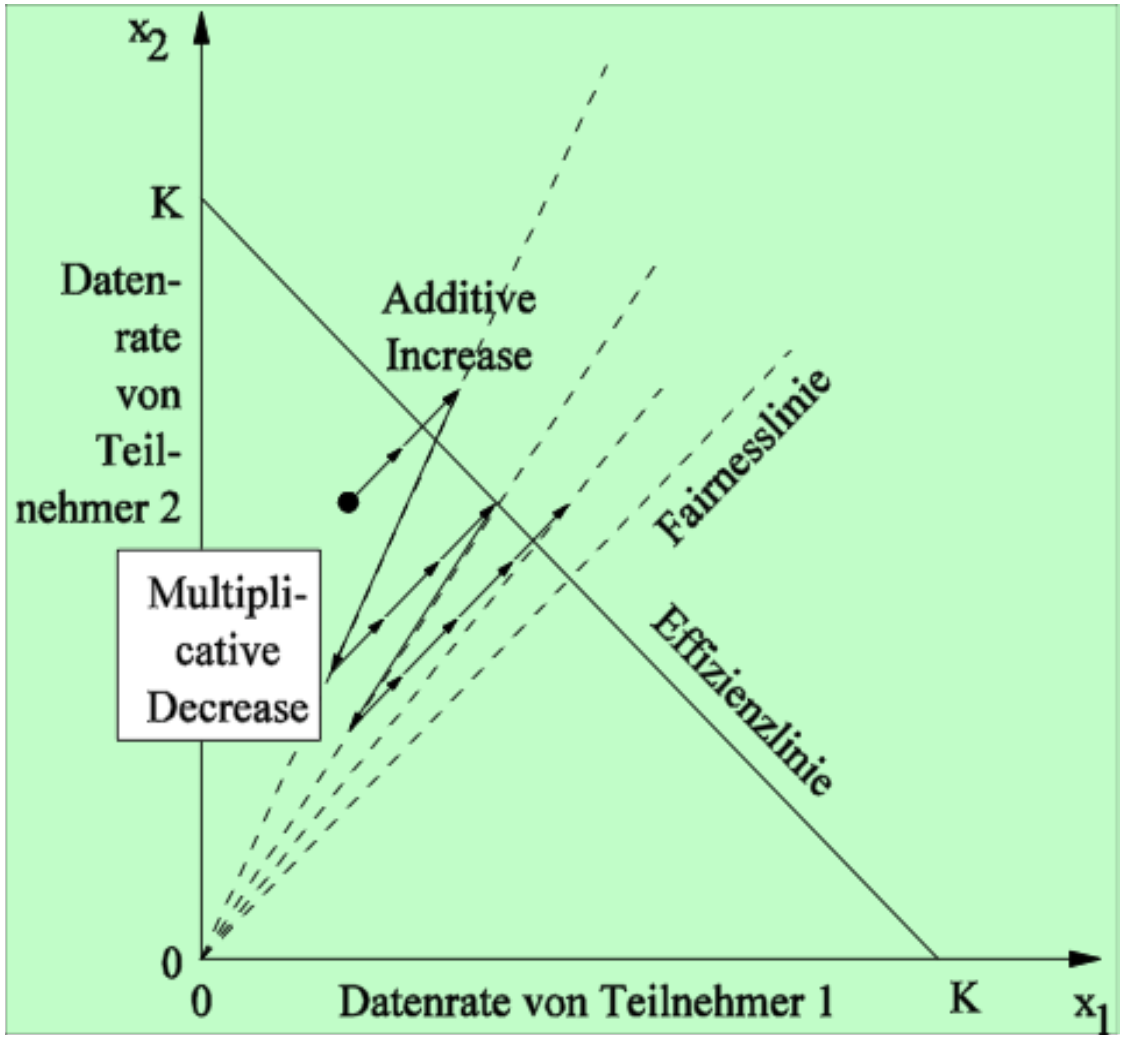


AIAD Additive Increase/ Additive Decrease

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer




x2
K
Multiplicative
Increase
Fairnesslinie
Daten-
rate
von
Teil-
nehmer 2
Multipli-
cative
Decrease
Effizienzlinie
0
0
Datenrate von Teilnehmer 1
K
x1


x2
K
Daten-
rate
von
Teil-
nehmer 2
Additive
Increase
Multipli-
cative
Decrease
Fairnesslinie
Effizienzlinie
0
0
Datenrate von Teilnehmer 1
K
x1

Ende der 4. Vorlesung

Peer-to-Peer-Netzwerke

Christian Schindelhauer

schindel@informatik.uni-freiburg.de

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer