Systeme II

5. Die Transportschicht

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg

Version 16.06.2014

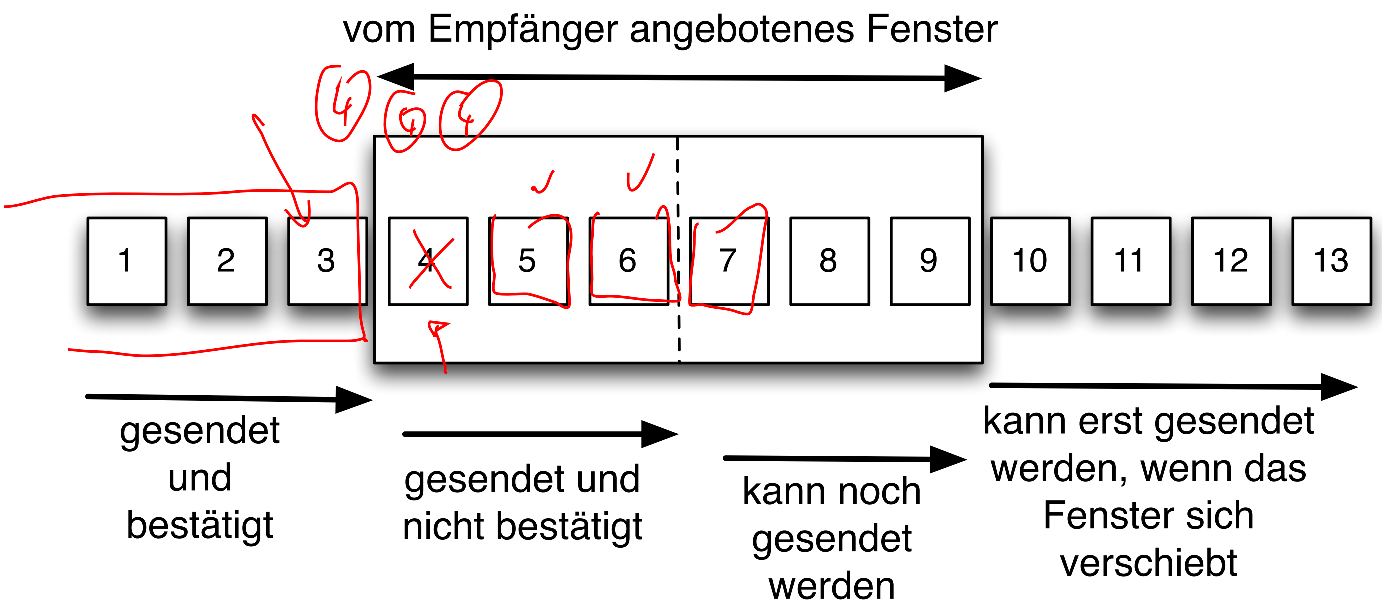
TCP-Header

- Sequenznummer
  - Nummer des ersten Bytes im Segment
  - Jedes Datenbyte ist nummeriert modulo 2^{32}
- Bestätigungsnummer
  - Aktiviert durch ACK-Flag
  - Nummer des nächsten noch nicht bearbeiteten Datenbytes
    - = letzte Sequenznummer + letzte Datenmenge:
- Port-Adressen
  - Für parallele TCP-Verbindungen
  - Ziel-Port-Nr.
  - Absender-Port
- Headerlänge
  - data offset
- Prüfsumme
  - Für Header und Daten

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

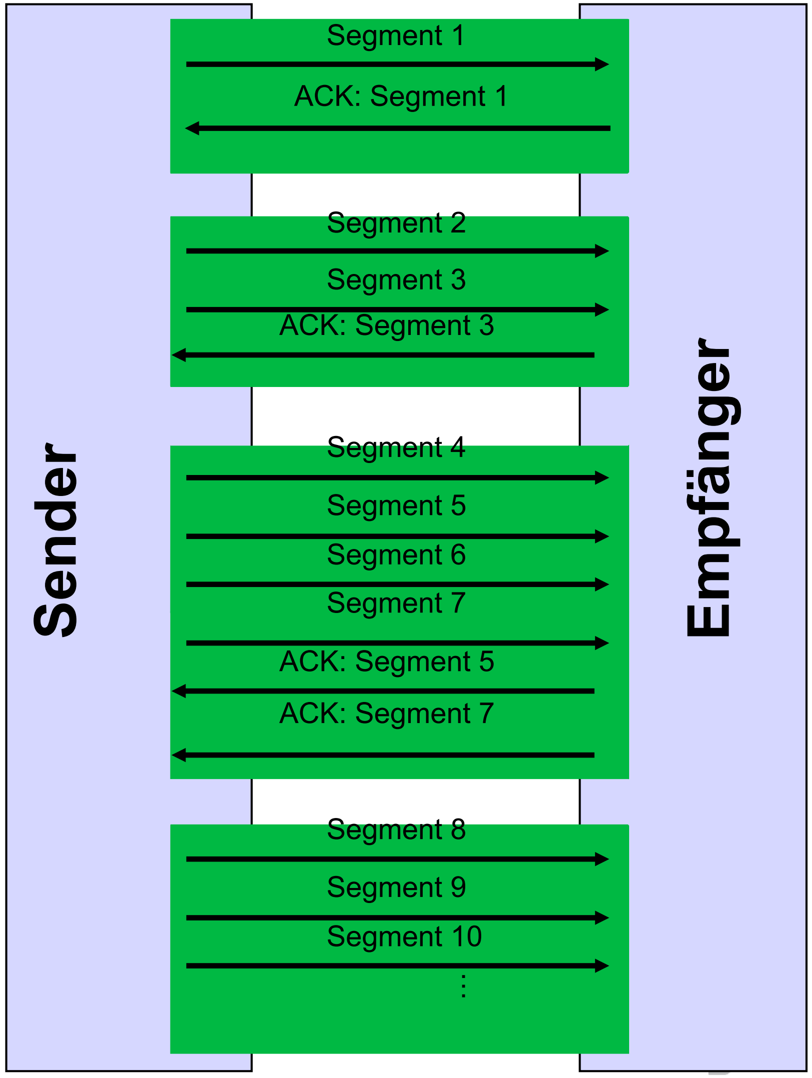
Gleitende Fenster (sliding windows)

- 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

- 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 $\leftarrow$ S
  - Für jede empfangene Bestätigung:
    - cwnd $\leftarrow$ cwnd + S
  - Solange bis einmal Bestätigung ausbleibt
- „Slow Start" = Exponentielles Wachstum



Stauvermeidung in TCP Tahoe

CoNe Freiburg

der Größe

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

$x \leftarrow 1$ | **$y \leftarrow$ max**

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

$x \leftarrow 1$ | **$y \leftarrow x/2$**

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

$x \leftarrow 2 \cdot x$, bis x = y

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

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

$x \leftarrow x + 1$

Bytes pro RTT = cwnd

Pakete = $\frac{cwnd}{S}$

$S \cdot \frac{S}{cwnd} \cdot \frac{cwnd}{S}$

UNI FREIBURG

TCP Tahoe

1kB = 1S = 1MSS

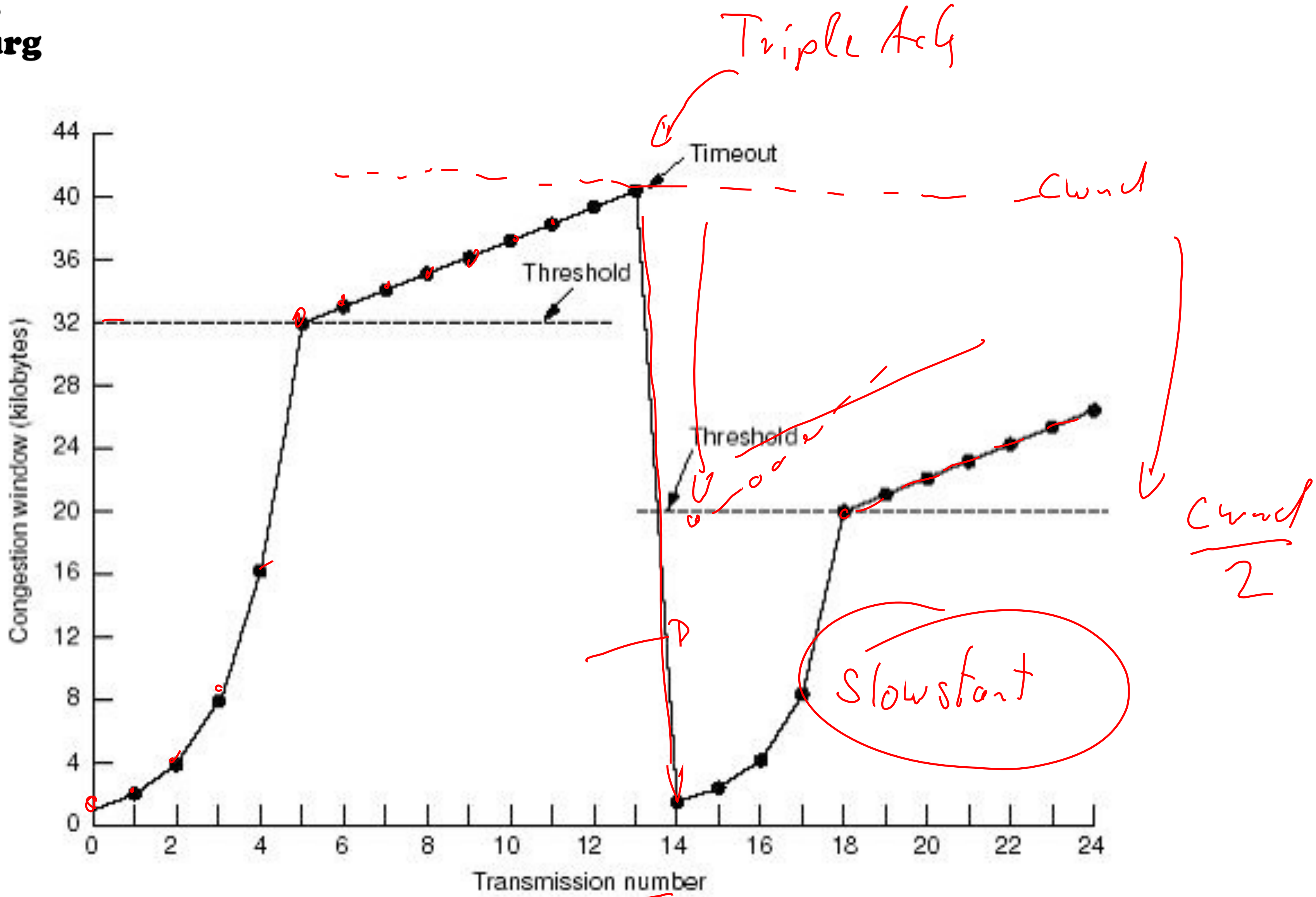


Fig3

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

Fast Retransmit und Fast Recovery

- 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 $\leftarrow$ min(wnd,cwnd)/2
    - cwnd $\leftarrow$ ssthresh + 3 S
  - Fast recovery nach Fast retransmit
    - Erhöhe Paketrate mit jeder weiteren Bestätigung
    - cwnd $\leftarrow$ cwnd + S
  - Congestion avoidance: Trifft Bestätigung von P+x ein:
    - cwnd $\leftarrow$ ssthresh

y $\leftarrow$ x/2
x $\leftarrow$ y + 3

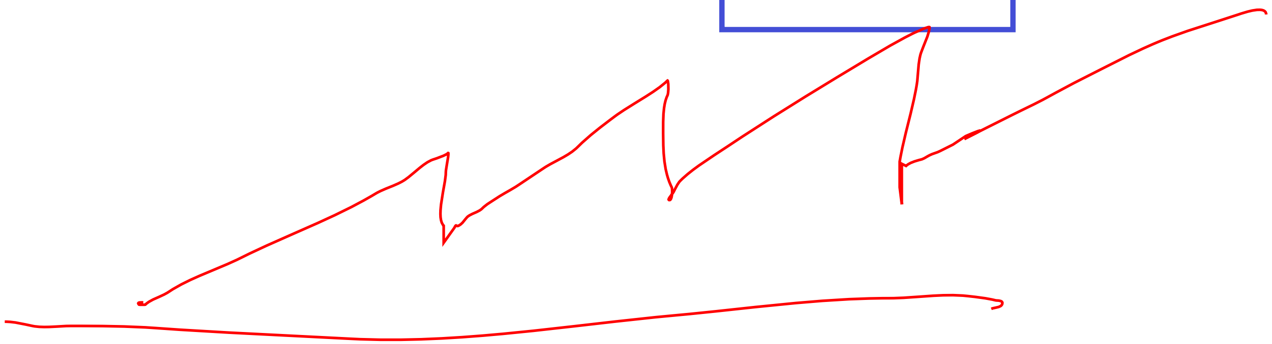
$x \leftarrow y$

Stauvermeidungsprinzip: AIMD

- Kombination von TCP und Fast Recovery verhält sich im wesentlichen wie folgt:
  - Verbindungsaufbau: $\boxed{x \leftarrow 1}$
  - Bei Paketverlust, MD:multiplicative decreasing

    $\boxed{x \leftarrow x/2}$
  - Werden Segmente bestätigt, AD: additive increasing

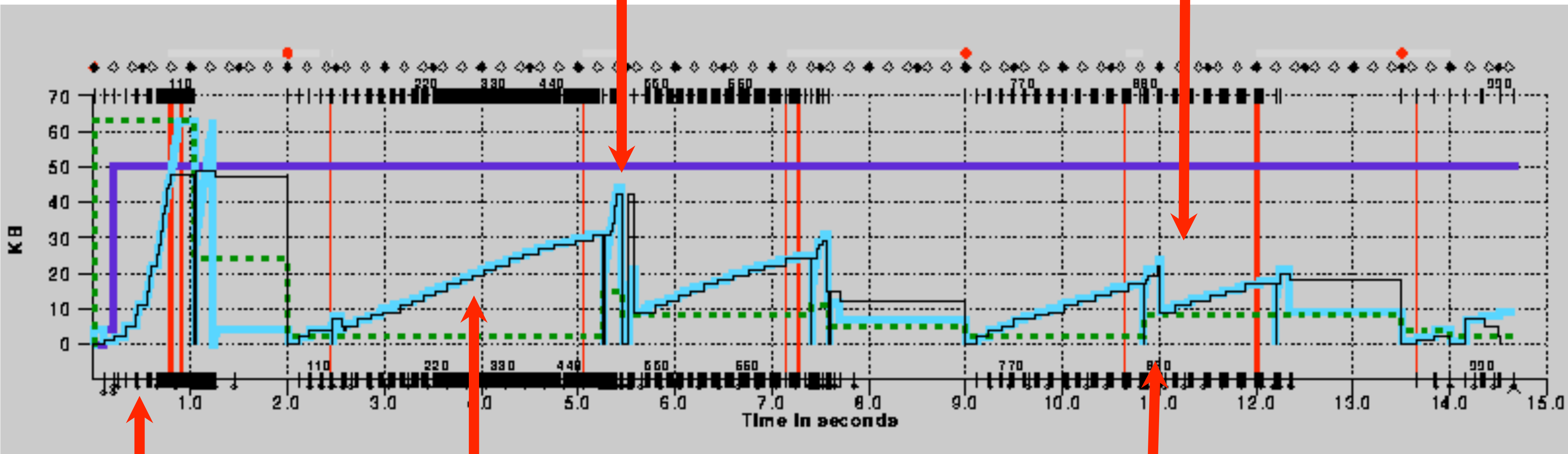
    $\boxed{x \leftarrow x+1}$



Beispiel: TCP Reno in Aktion

Fast Retransmit

Fast Recovery



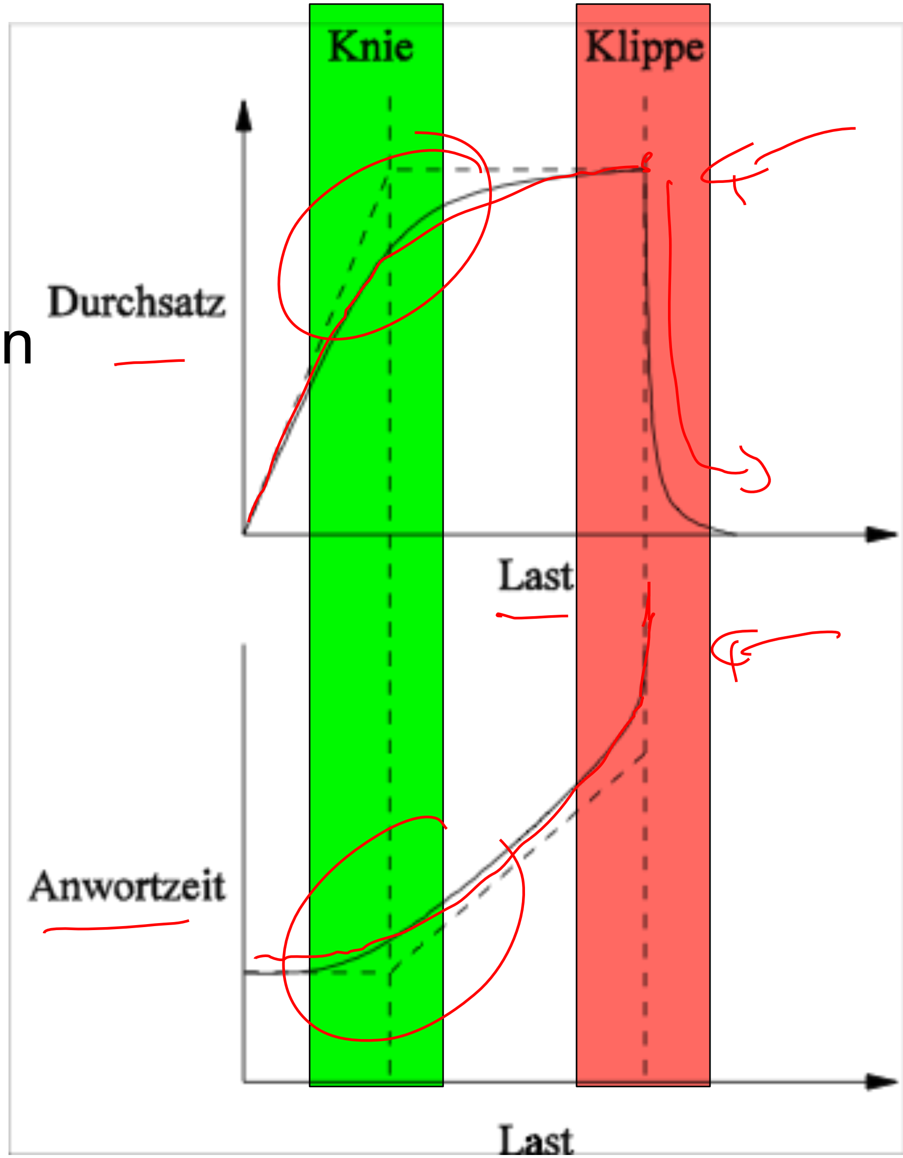
Additively Increase

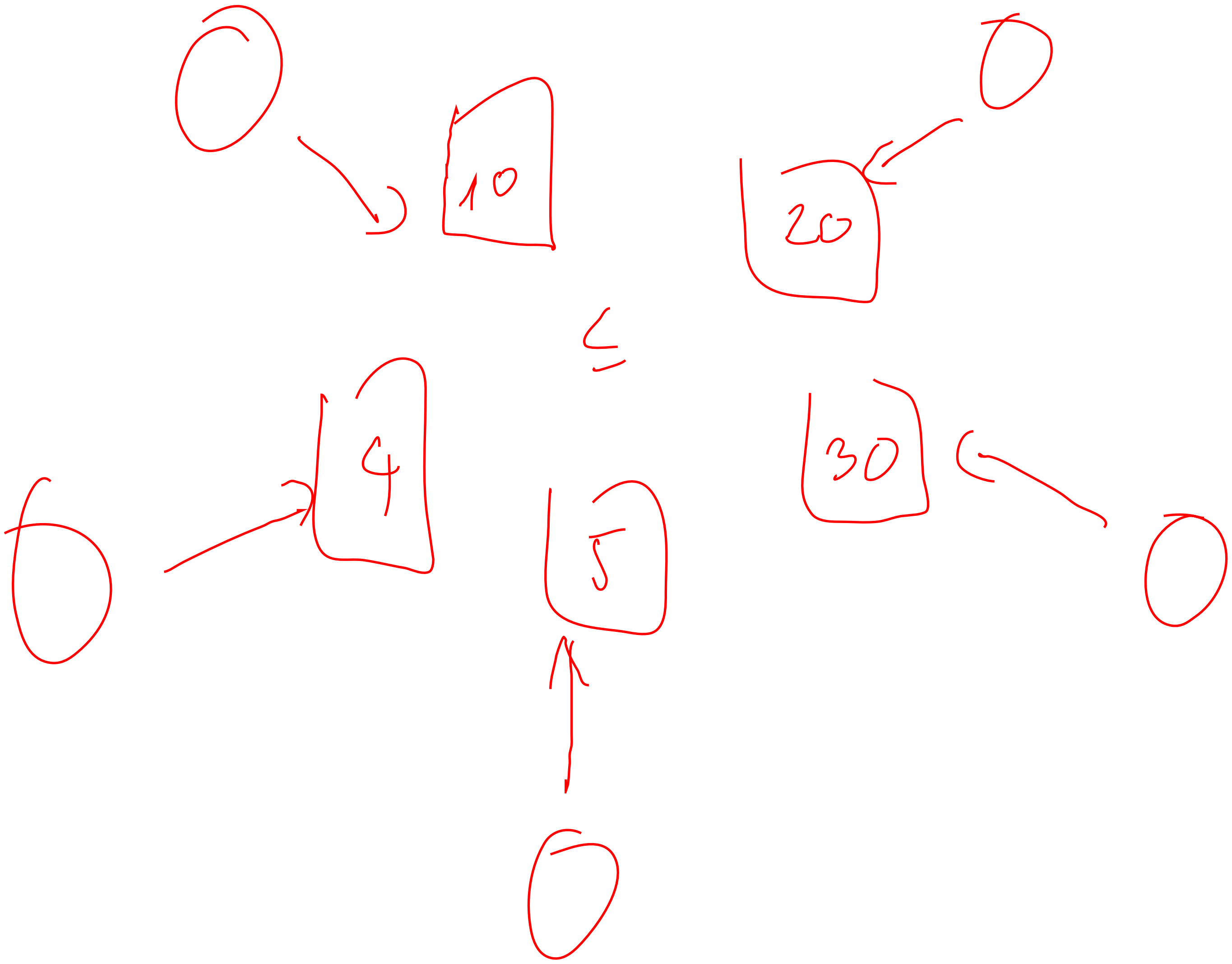
Multiplicatively Decrease

Slow Start

Durchsatz und Antwortzeit

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




0
10
0
20
≤
4
5
30
0
0
0

CoNe Freiburg

Ein einfaches Datenratenmodell

$x_1(0)$ $x_1(1)$ $x_1(2)$

10 11 12 13

- n Teilnehmer, Rundenmodell
  - Teilnehmer i hat Datenrate $x_i(t)$
  - 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:

$2x$ $\qquad\qquad$ $\sim 1 + x$

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

UNI FREIBURG

Lineare Datenratenanpassung

- Interessante Spezialfälle:

$x+1$ $x-1$

  - 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 $a_D < 0$.

$2x$ $x/2$

  - 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 $b_D < 1$.

$x+1$ $x/2$

  - 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 $b_D < 1$.

MIAD $2x$ $x-1$

CoNe Freiburg

Fairness und Effizienz

- Effizienz
  - Last: $X(t) := \sum_{i=1}^{n} x_i(t)$
  - Maß $|X(t) - K|$
- Fairness: Für x=(x1, …, xn):

$$F(x) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} (x_i)^2} .$$

  - 1/n ≤ F(x) ≤ 1
  - F(x) = 1 ↔ absolute Fairness ✓
  - Skalierungsunabhängig ✓
  - Kontinuierlich, stetig, differenzierbar ✓
  - Falls k von n fair, Rest 0, dann F(x) = k/n

$(1, 0)$

$\frac{1}{2 \cdot 1} = \frac{1}{2}$

$(1, 1)$

$\frac{4}{2 \cdot 2} = 1$

$(2, 1)$

$\frac{9}{2 \cdot 5} = \frac{9}{10}$

$\Downarrow$

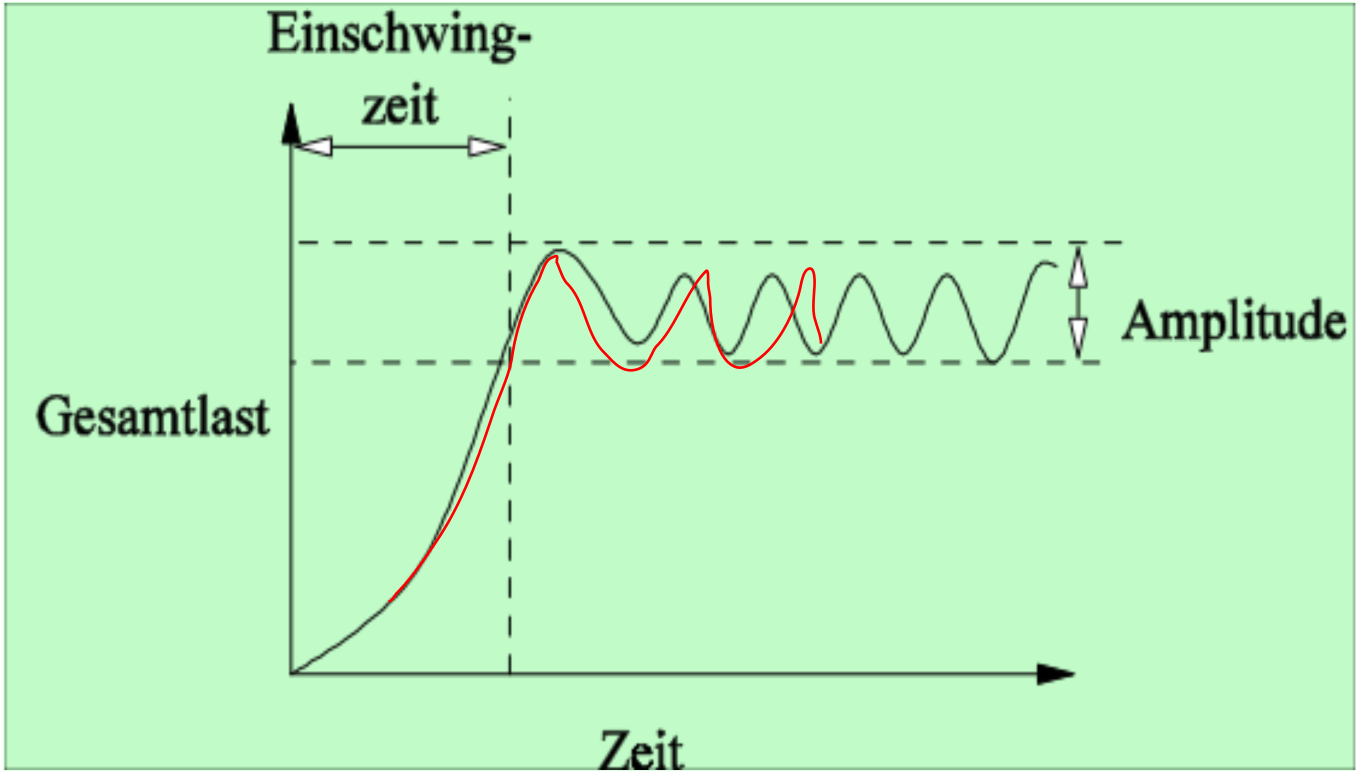
$(4, 2)$

$\frac{36}{2 \cdot 20} = \frac{9}{10}$

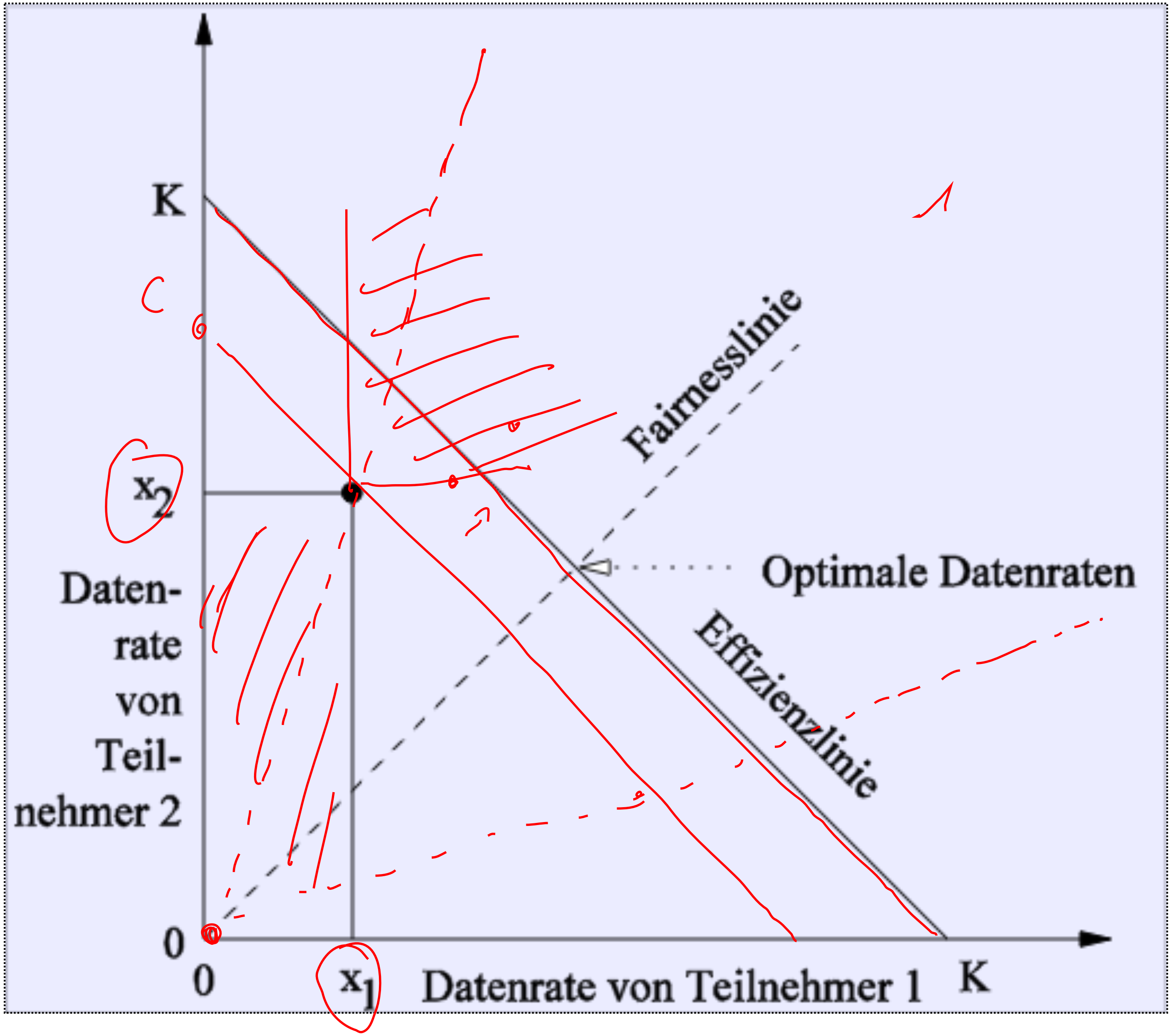
UNI FREIBURG

Konvergenz

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



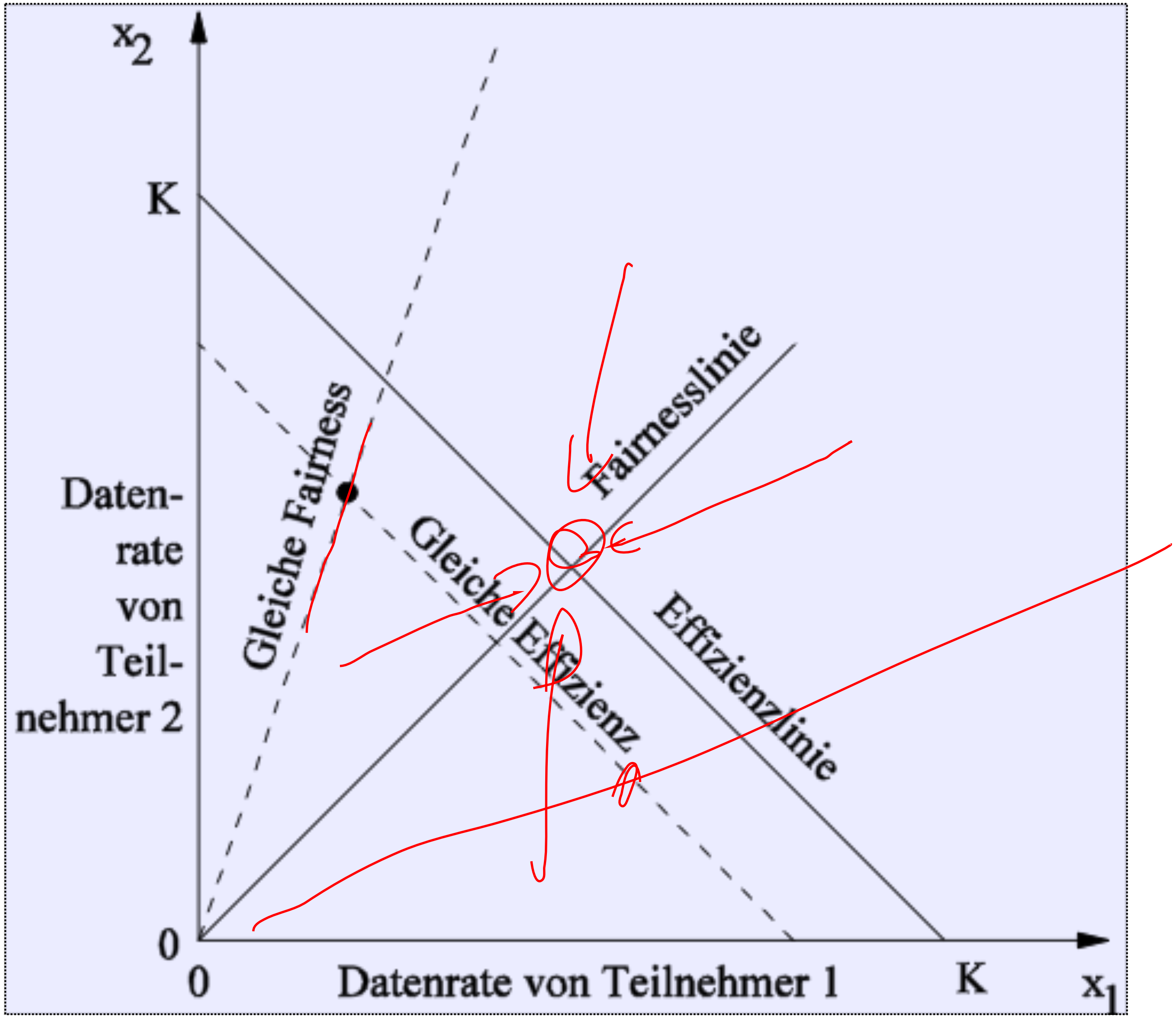
Vektordarstellung (I)



$x_1 + x_2 = C$

$x_2 = C - x_1$

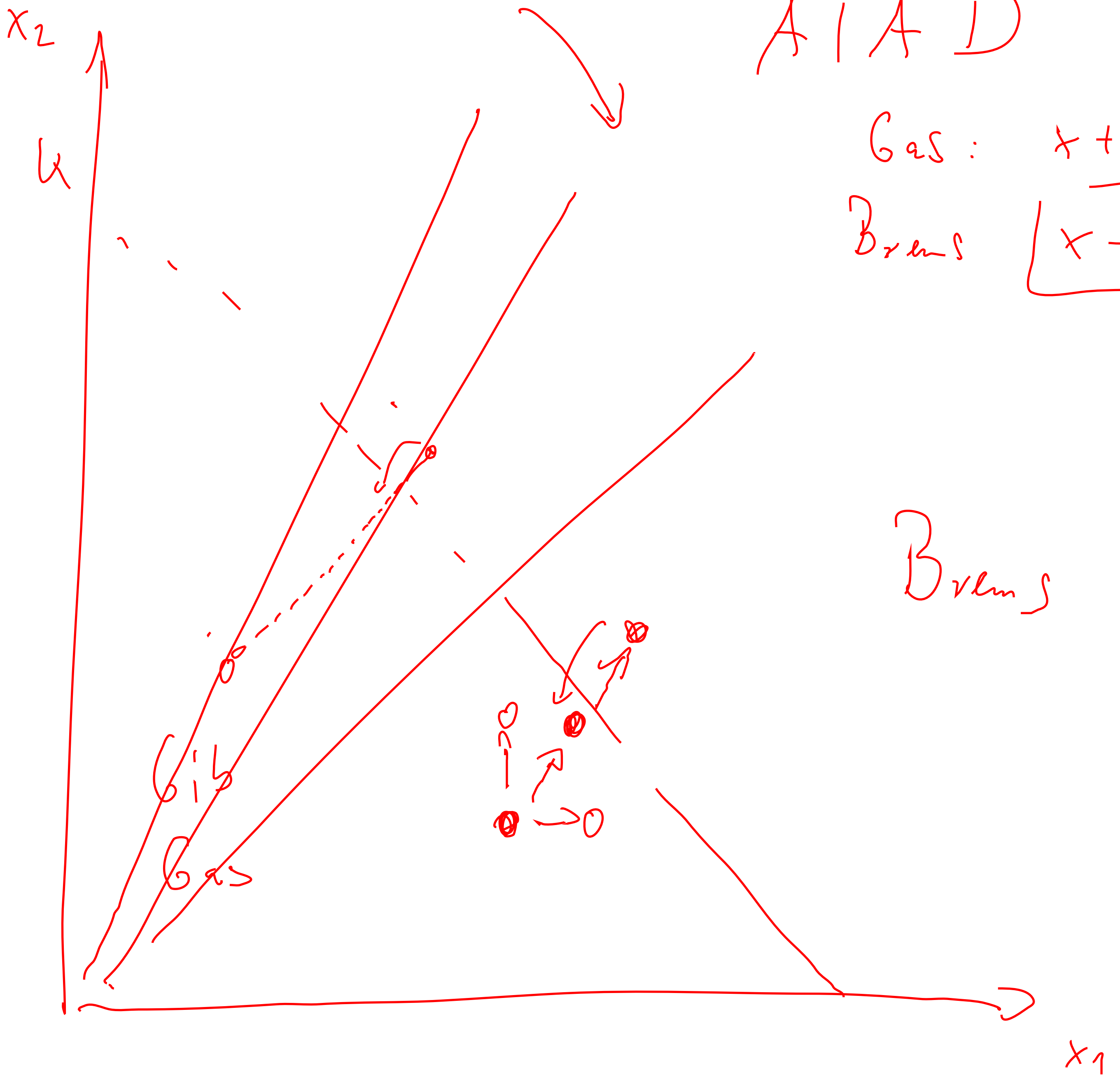
Vektordarstellung (II)



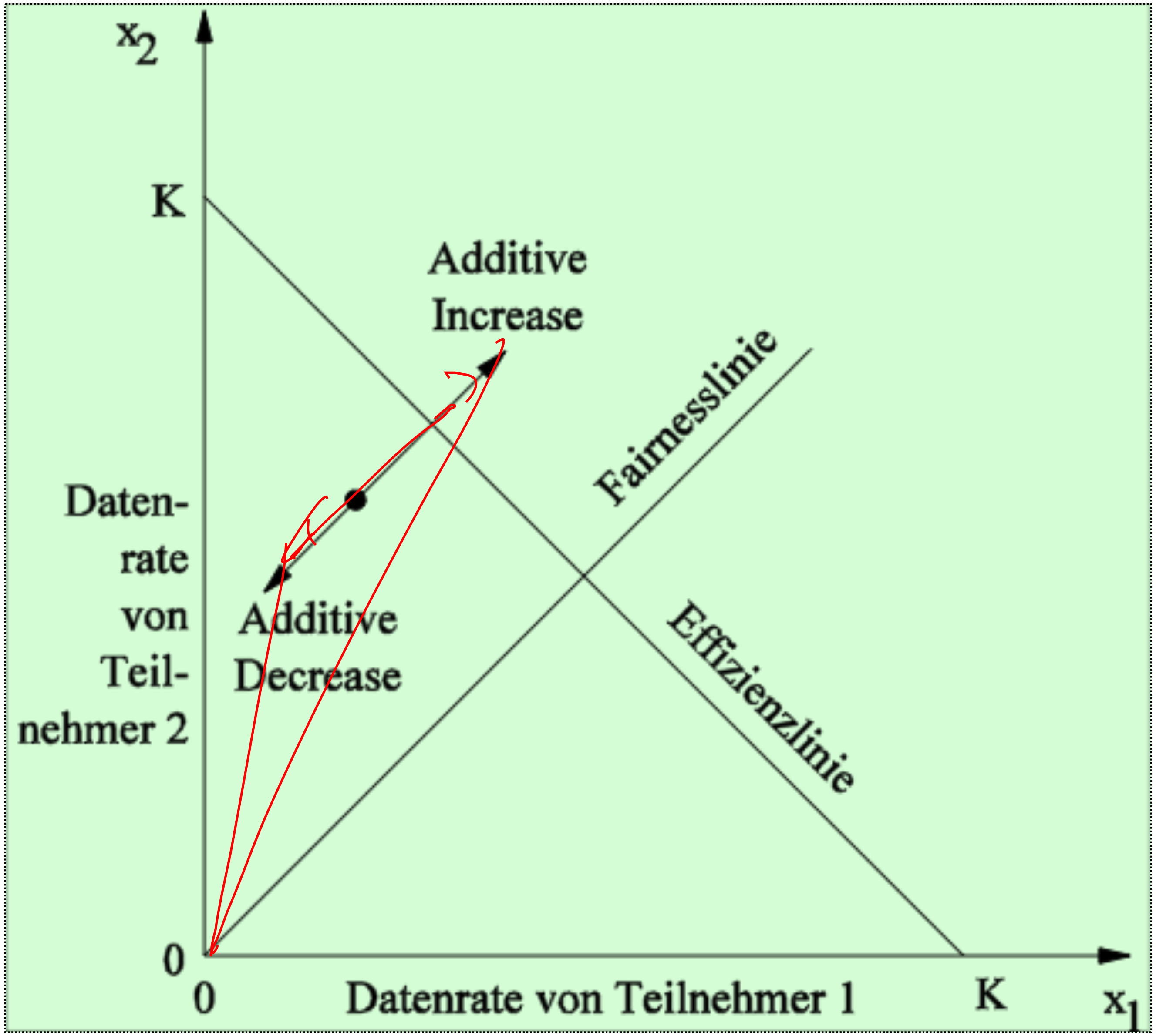
AIAD

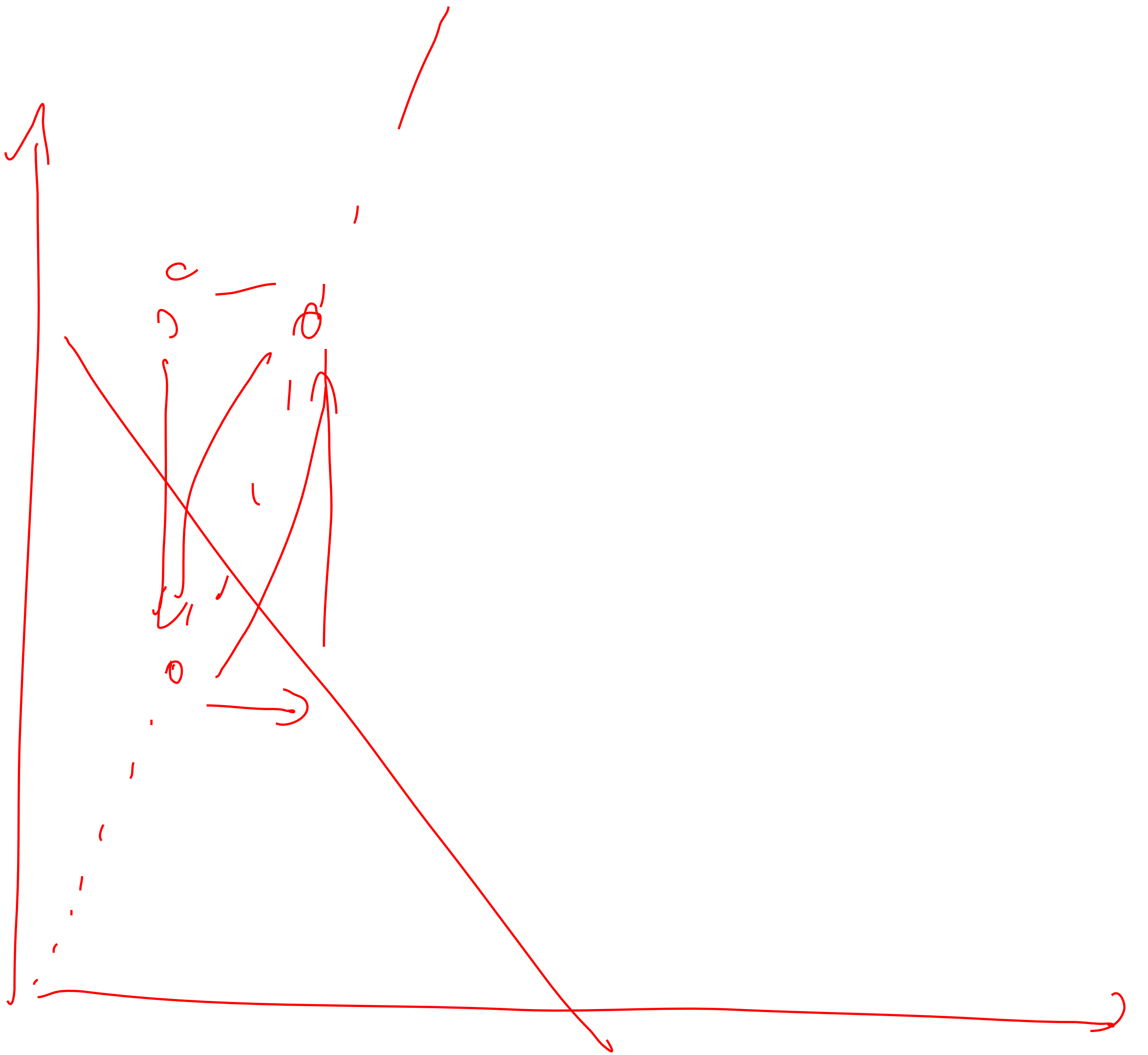
Gas: $x+1$

Brems $\boxed{x-1}$

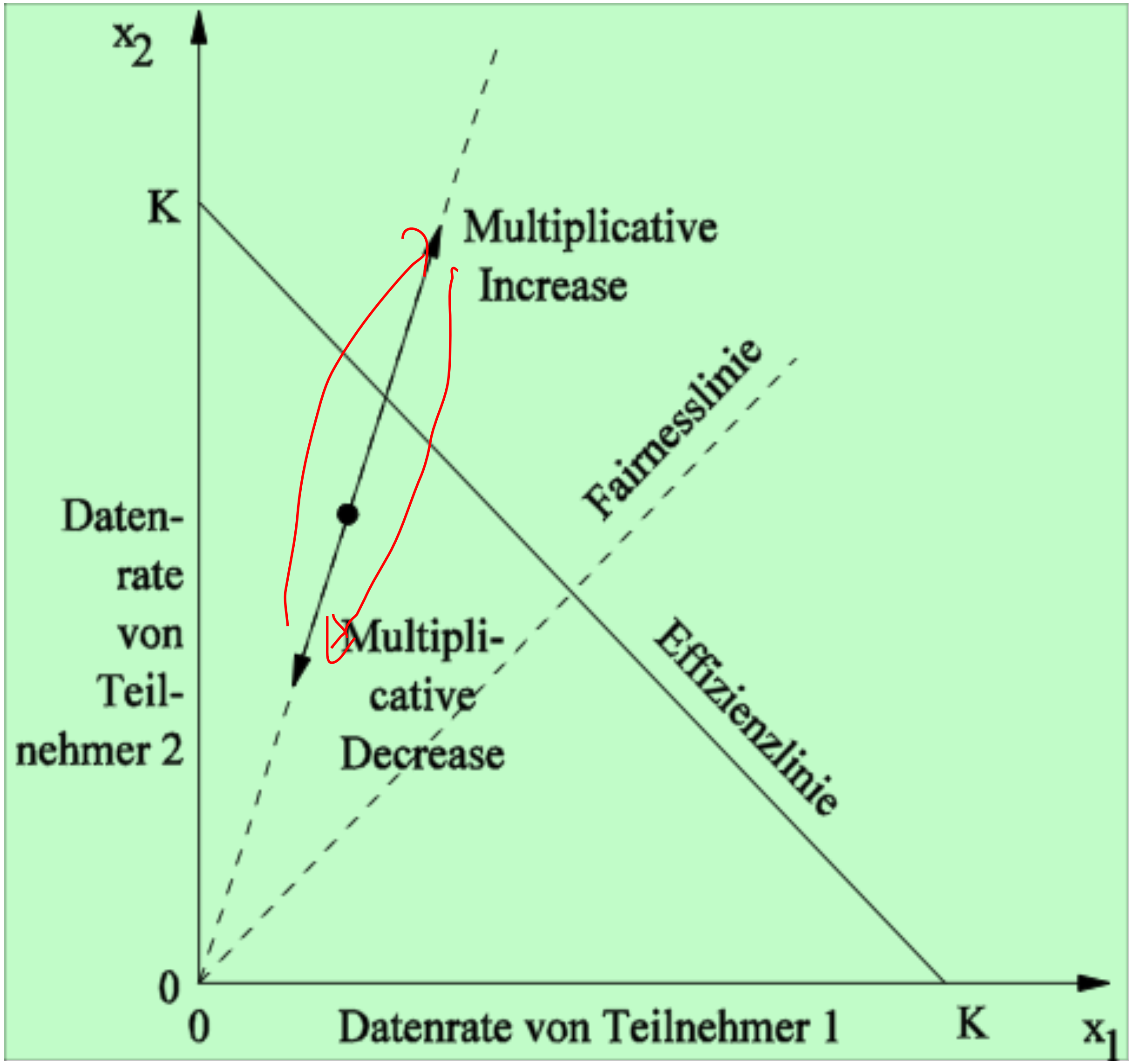


AIAD Additive Increase/ Additive Decrease

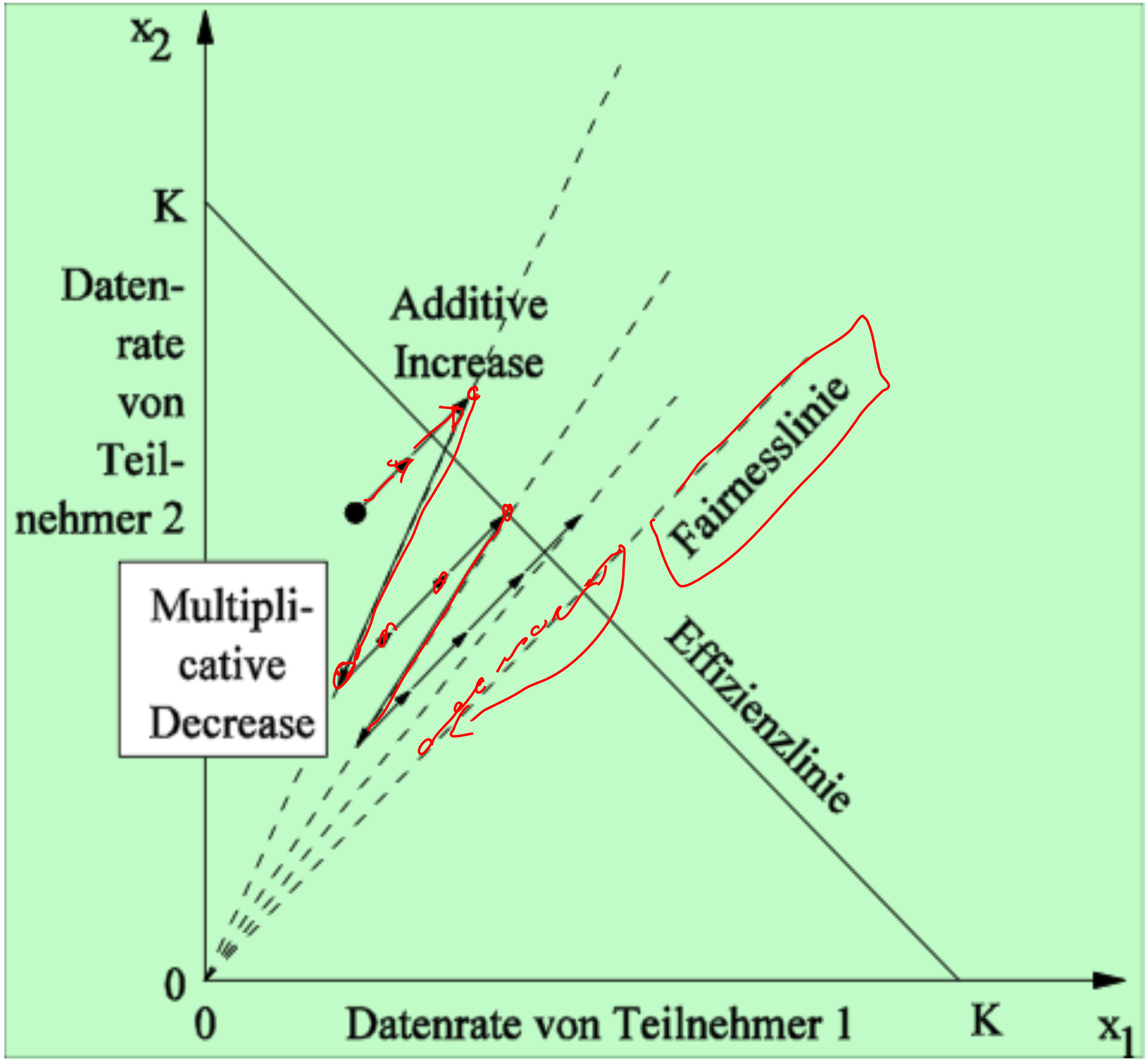




MIMD: Multiplicative Incr./ Multiplicative Decrease



AIMD: Additively Increase/ Multiplicatively Decrease



Systeme II

5. Die Transportschicht

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg