

Systeme II



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

Christian Schindelhauer

Sommersemester 2006

19. Vorlesung

12.07.2006

schindel@informatik.uni-freiburg.de



Stauvermeidung in TCP Tahoe

x: Anzahl Pakete pro RTT

➤ **Jacobson 88:**

- Parameter: cwnd und Slow-Start-Schwellwert (ssthresh=slow start threshold)
- S = Datensegmentgröße = maximale Segmentgröße

➤ **Verbindungsaufbau:**

$$cwnd \leftarrow S$$

$$ssthresh \leftarrow 65535$$

$$x \leftarrow 1$$

$$y \leftarrow \max$$

➤ **Bei Paketverlust, d.h. Bestätigungsdauer > RTO,**

- multiplicatively decreasing

$$cwnd \leftarrow S$$

$$ssthresh \leftarrow \max \left\{ 2 \times S, \frac{\min \{ cwnd, wnd \}}{2} \right\}$$

$$x \leftarrow 1$$

$$y \leftarrow x/2$$

➤ **Werden Segmente bestätigt und $cwnd \leq ssthresh$, dann**

- slow start:

$$cwnd \leftarrow cwnd + S$$

$$x \leftarrow 2 \cdot x, \text{ bis } x = y$$

➤ **Werden Segmente bestätigt und $cwnd > ssthresh$, dann additively**

increasing

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

$$x \leftarrow x + 1$$



TCP Tahoe

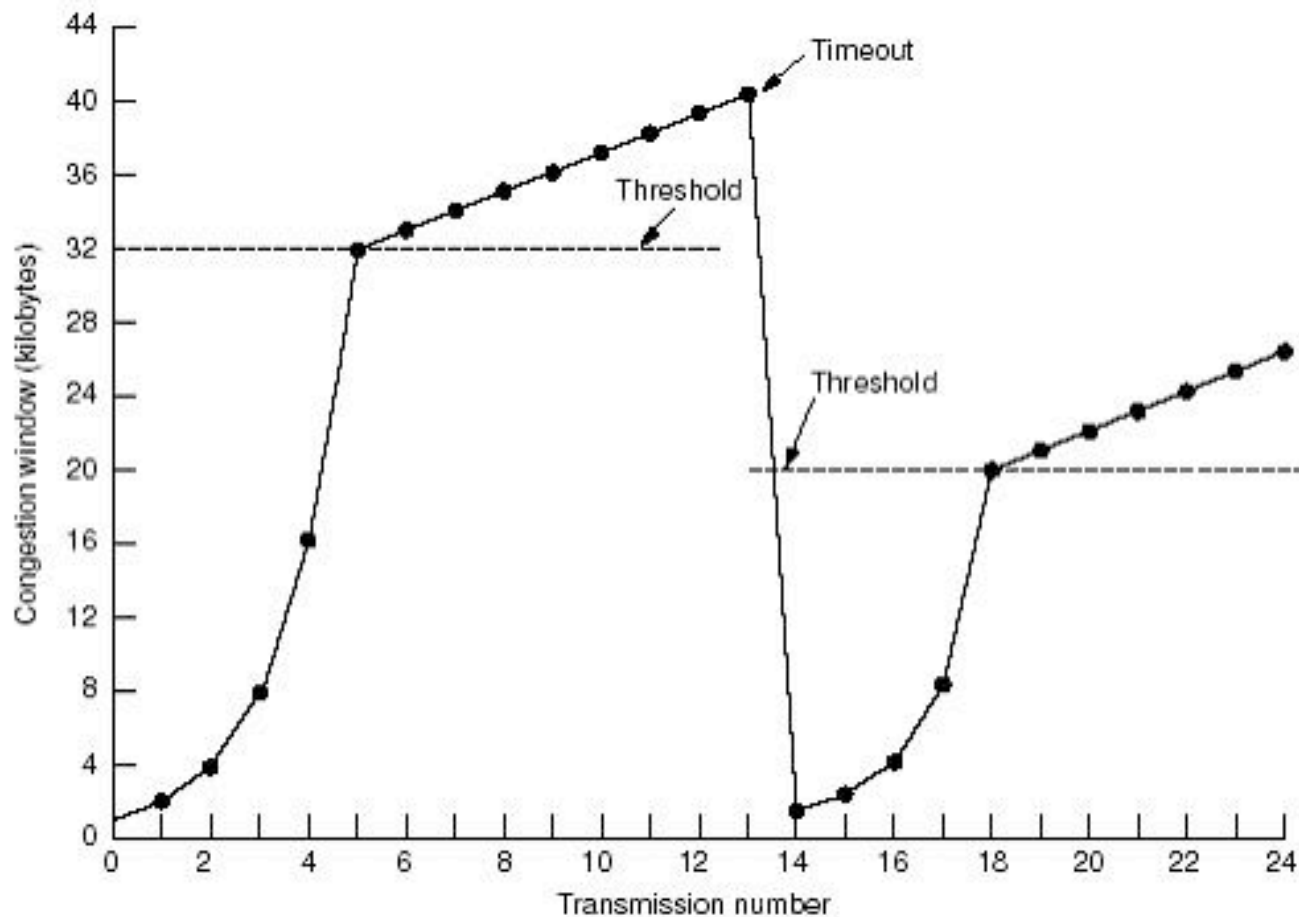


Fig3

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:
 - $cwnd \leftarrow ssthresh$

$$y \leftarrow x/2$$

$$x \leftarrow y + 3$$



Stauvermeidungsprinzip: AIMD

➤ 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

$$x \leftarrow x + 1$$

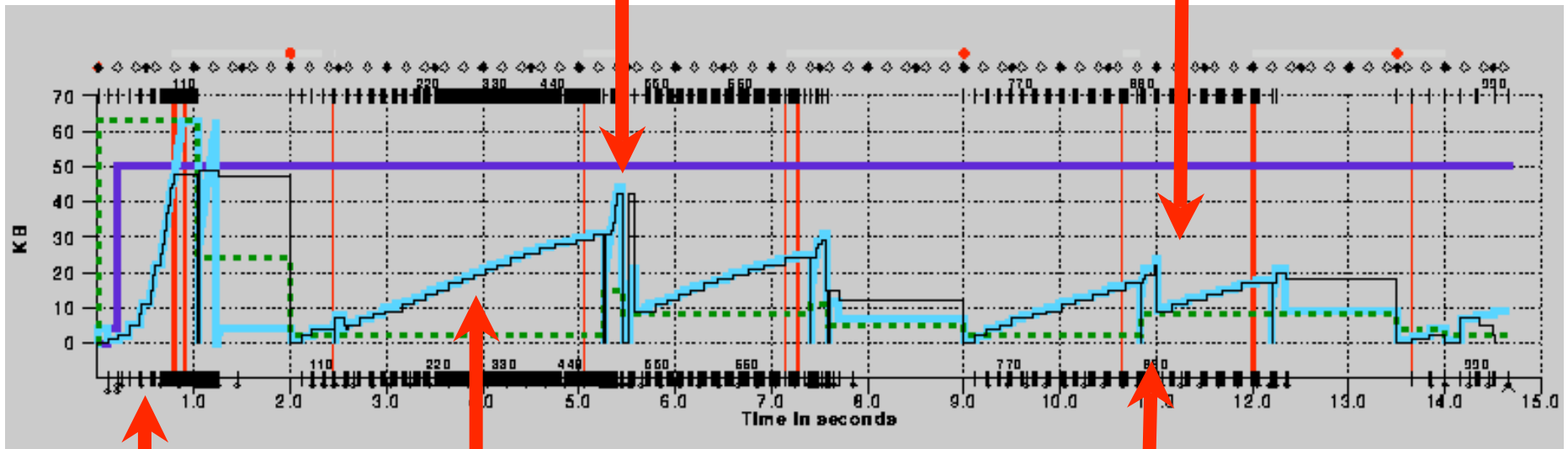


Beispiel: TCP Reno in Aktion

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

Fast Retransmit

Fast Recovery



Slow Start

Additively Increase

Multiplicatively Decrease



Durchsatz und Antwortzeit

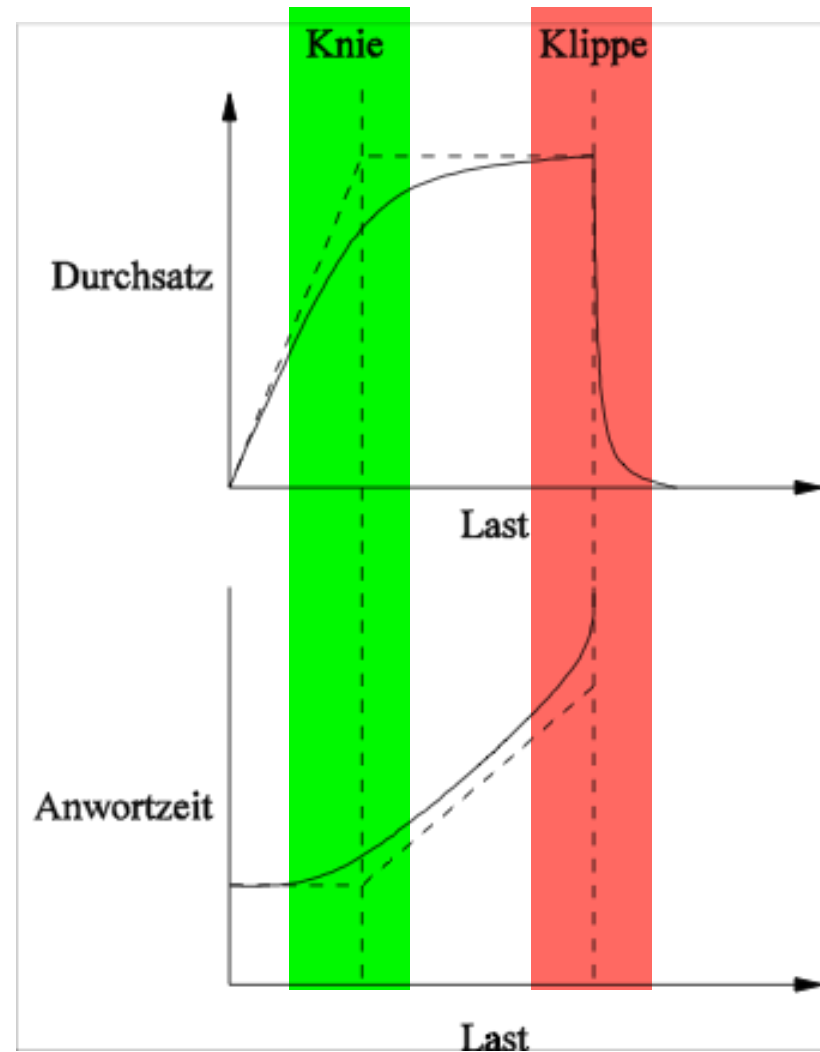
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

➤ **Klippe:**

- Hohe Last
- Geringer Durchsatz
- Praktisch alle Daten gehen verloren

➤ **Knie:**

- Hohe Last
- Hoher Durchsatz
- Einzelne Daten gehen verloren





Ein einfaches Datenratenmodell

➤ n Teilnehmer, Rundenmodell

- Teilnehmer i hat Datenrate $x_i(t)$
- Anfangsdatenrate $x_1(0), \dots, x_n(0)$ 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:

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



Lineare Datenratenanpassung

➤ 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

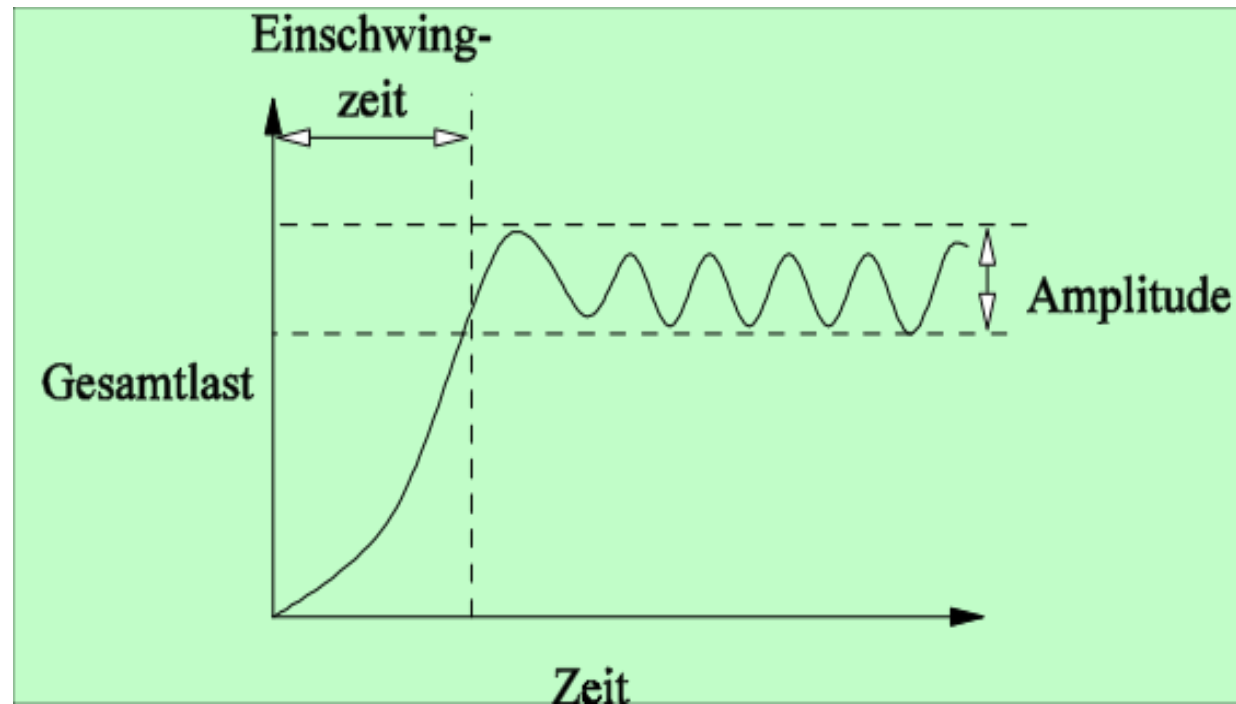
$$f_0(x) = a_I + x \quad \text{und} \quad f_1(x) = b_D x ,$$

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



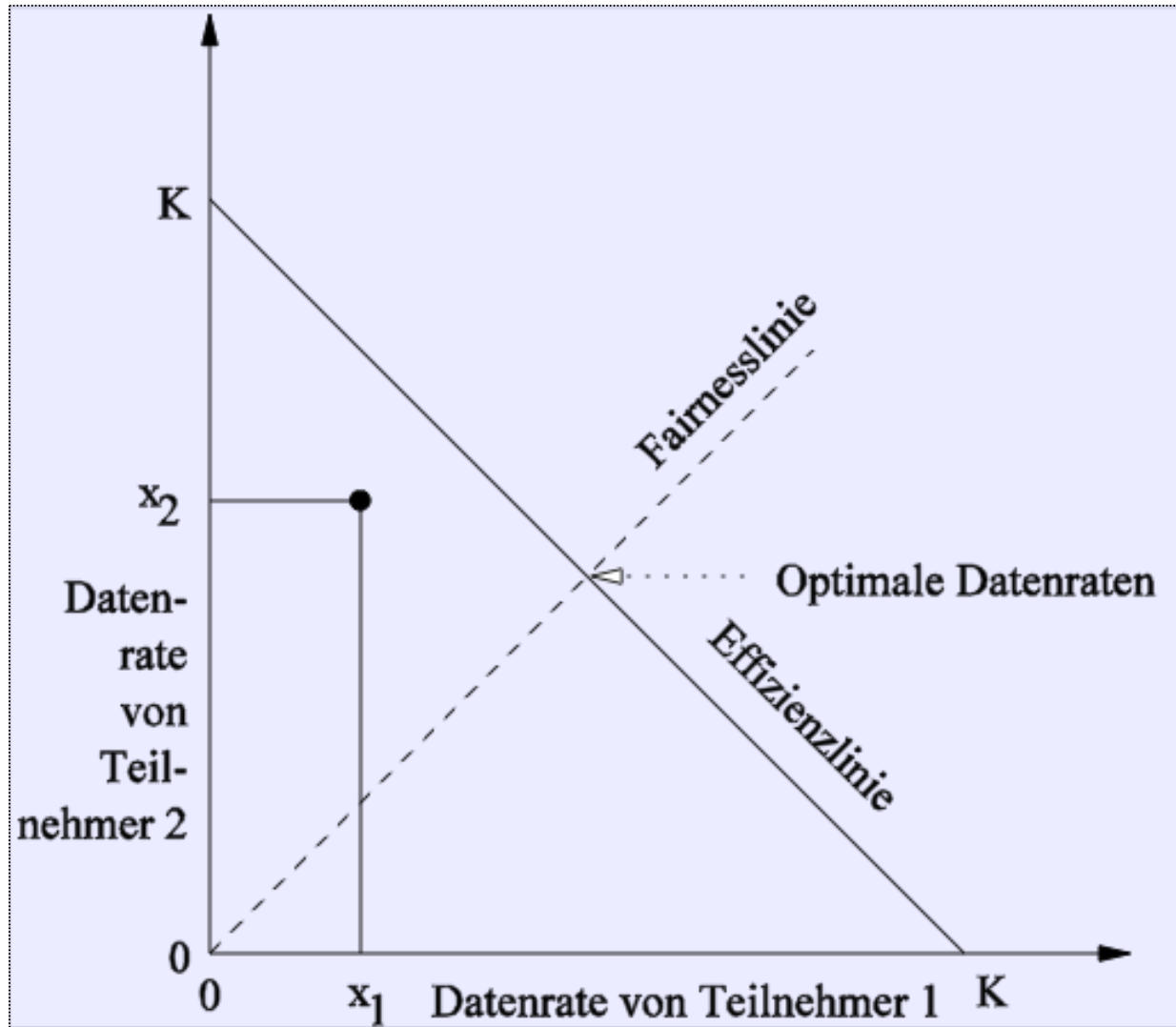
Konvergenz

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



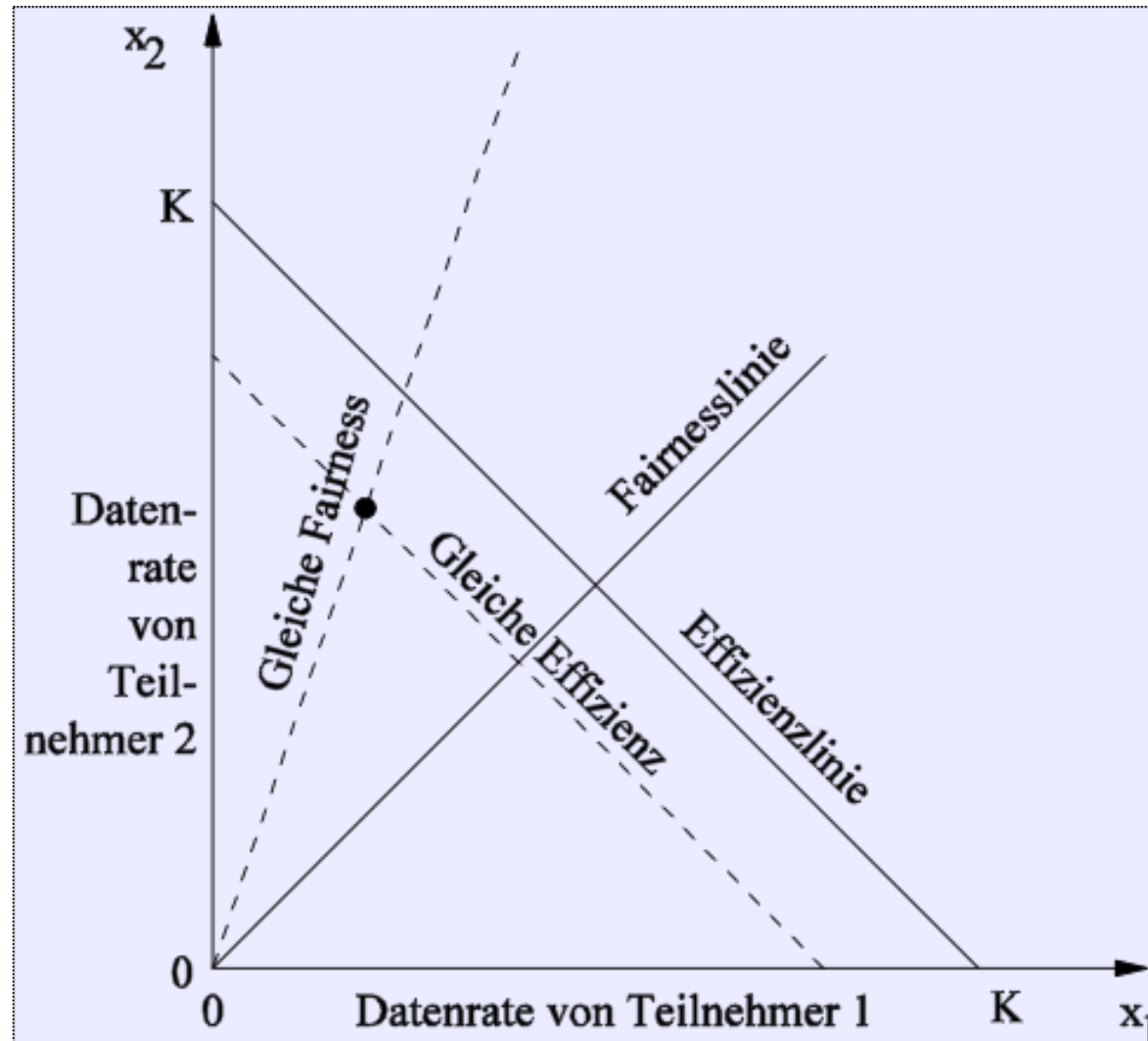


Vektordarstellung (I)





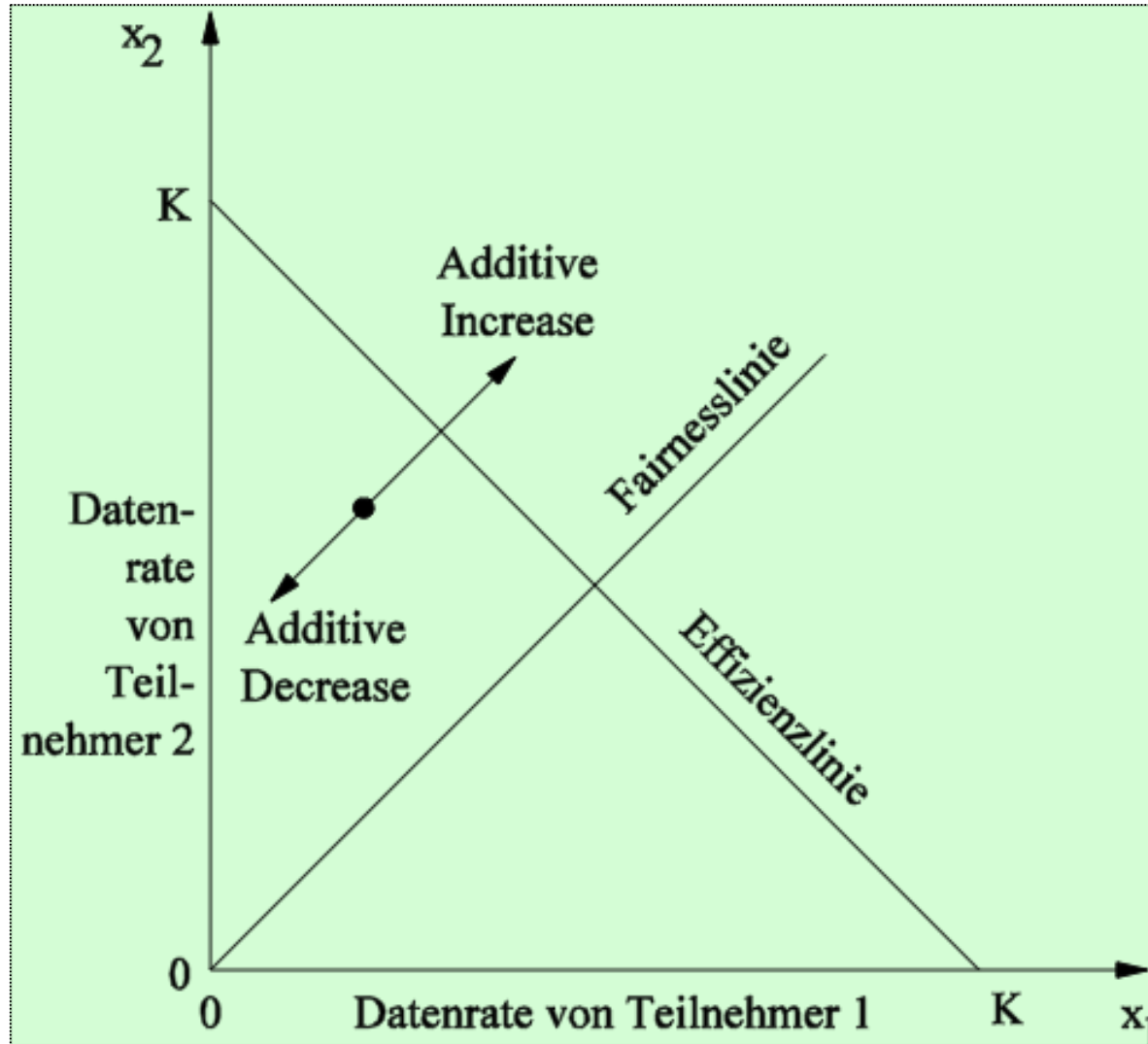
Vektordarstellung (II)





AIAD Additive Increase/ Additive Decrease

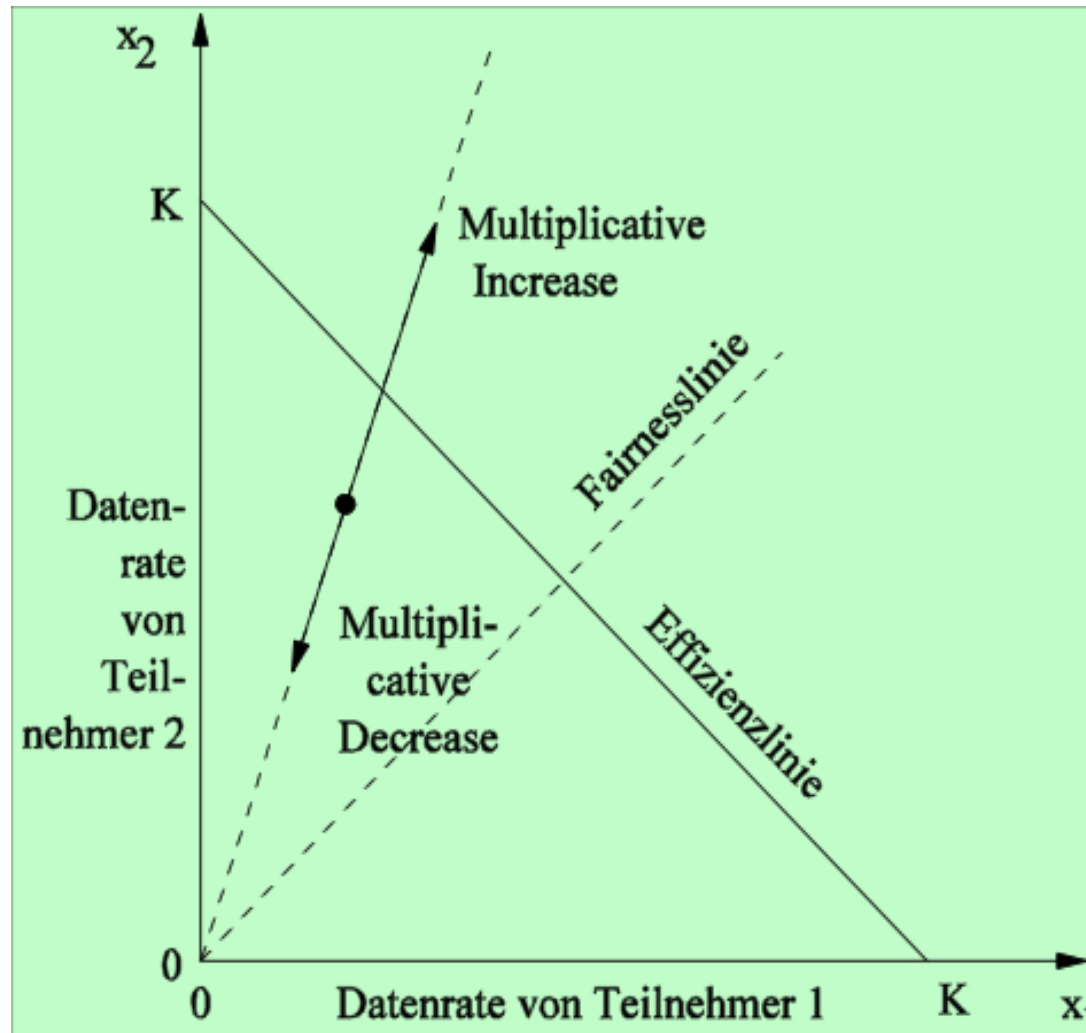
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer





MIMD: Multiplicative Incr./ Multiplicative Decrease

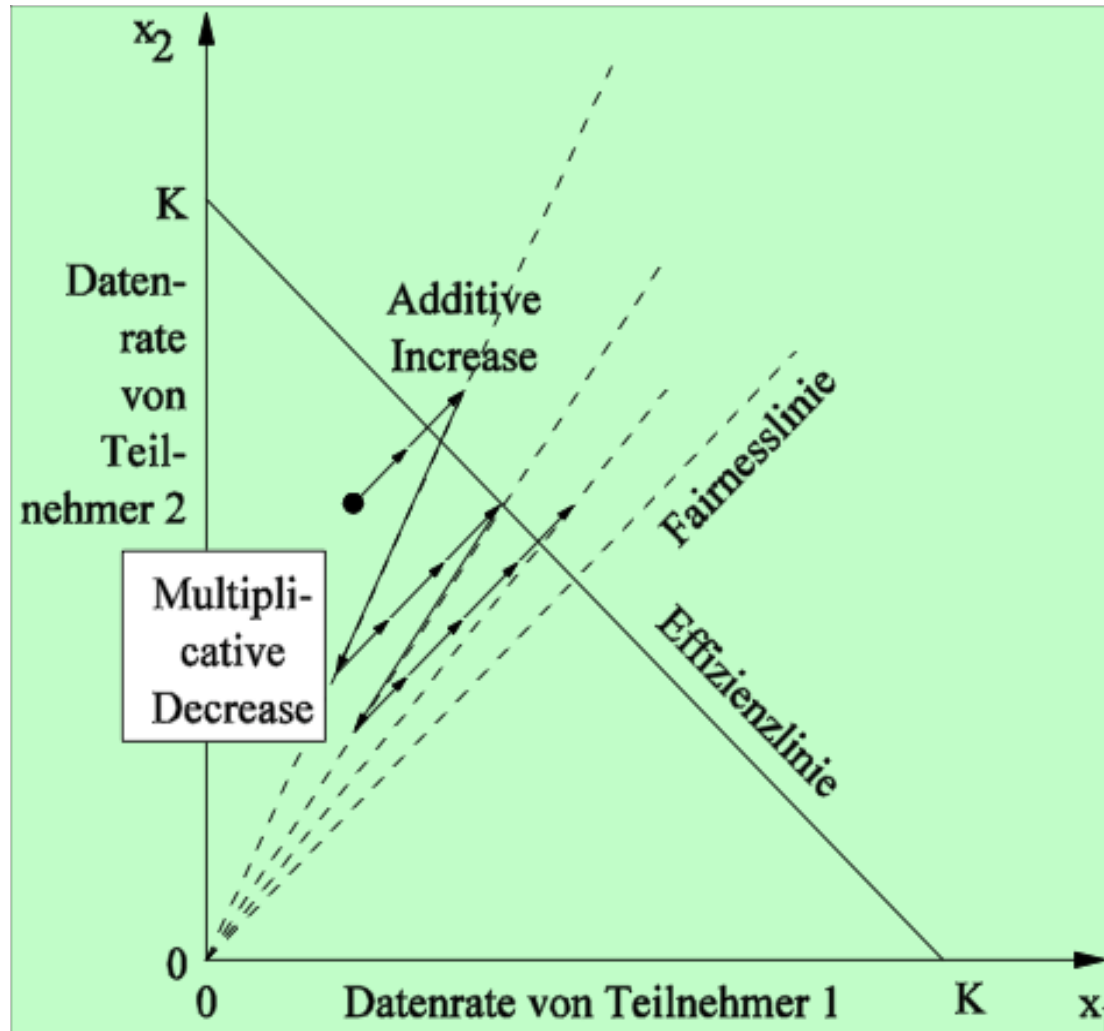
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer





AIMD: Additively Increase/ Multiplicatively Decrease

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





TCP

Zusammenfassung

- **TCP erzeugt zuverlässigen Byte-Strom**
 - Fehlerkontrolle durch “GoBack-N”
- **Congestion control**
 - Fensterbasiert
 - AIMD, Slow start, *Congestion Threshold*
 - Flusskontrolle durch *Window*
 - Verbindungsaufbau
 - Algorithmus von Nagle



TCP fairness & TCP friendliness

➤ TCP

- reagiert dynamisch auf die zur Verfügung stehende Bandweite
- Faire Aufteilung der Bandweite
 - Im Idealfall: n TCP-Verbindungen erhalten einen Anteil von $1/n$

➤ Zusammenspiel mit anderen Protokollen

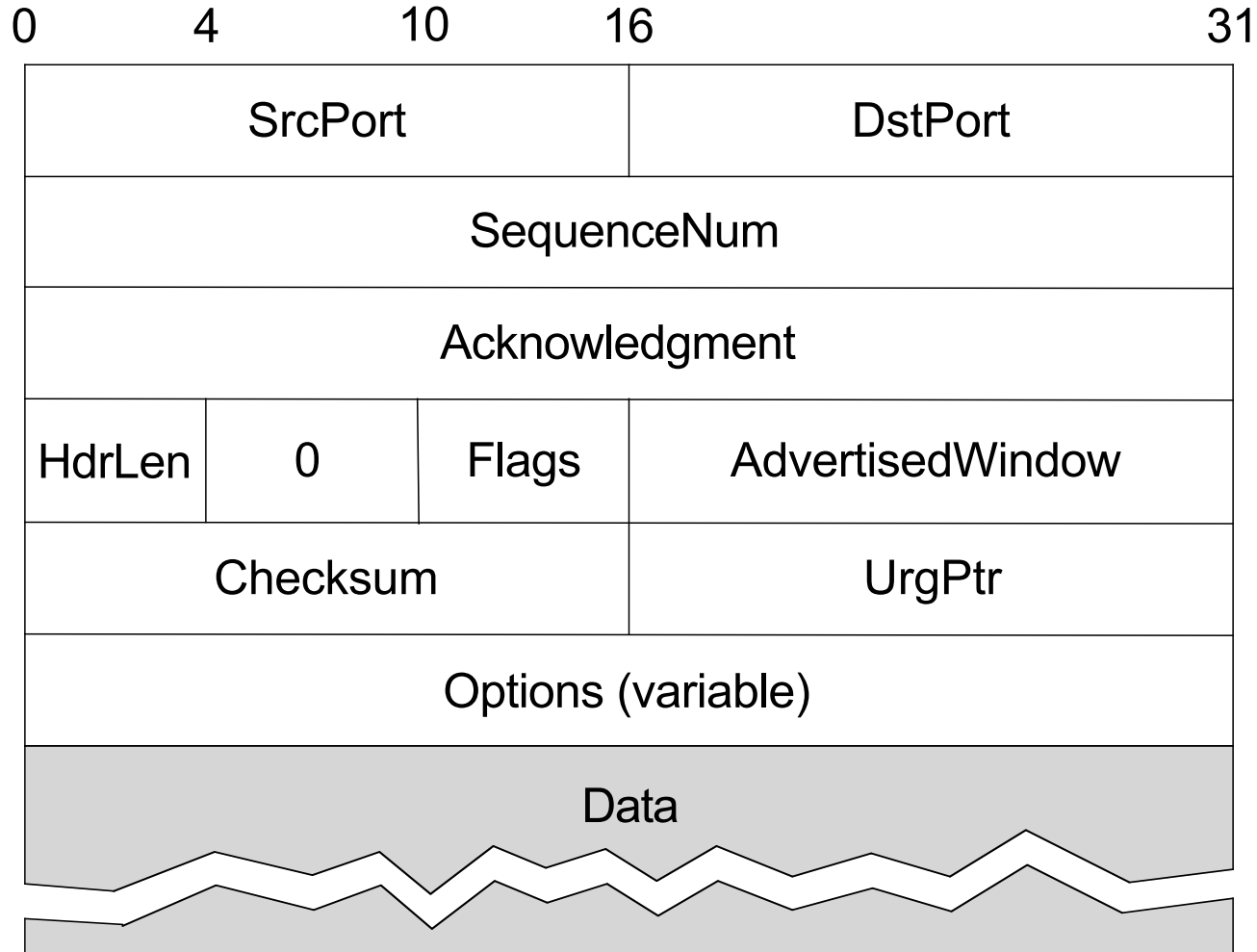
- Reaktion hängt von der Last anderer Transportprotokolle ab
 - z.B. UDP hat keine Congestion Control
- Andere Protokolle können jeder Zeit eingesetzt werden
- UDP und andere Protokoll können TCP Verbindungen unterdrücken

➤ Schlussfolgerung

- Transport-Protokolle müssen TCP-kompatibel sein (TCP friendly)



TCP header





Dienste der Transport-Schicht

- **Verbindungslos oder Verbindungsorientert**
 - Beachte: Sitzungsschicht im ISO/OSI-Protokoll
- **Verlässlich oder Unverlässlich**
 - Best effort oder Quality of Service
 - Fehlerkontrolle
- **Mit oder ohne Congestion Control**
- **Möglichkeit verschiedener Punkt-zu-Punktverbindungen**
 - Stichwort: Demultiplexen
- **Interaktionsmodelle**
 - Byte-Strom, Nachrichten, „Remote Procedure Call“



UDP

➤ User Datagram Protocol (UDP)

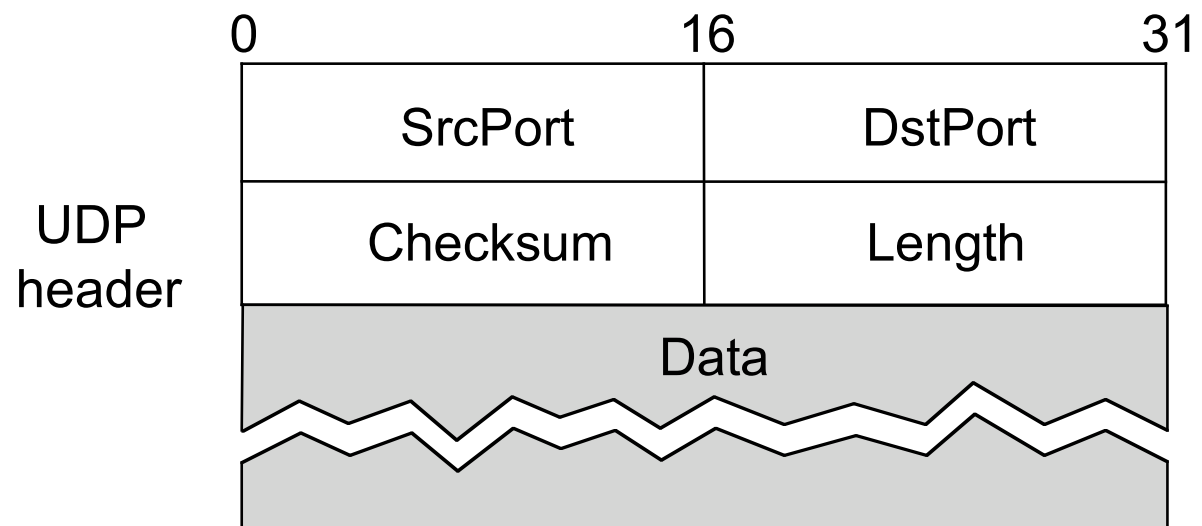
– ist ein unzuverlässiges, verbindungsloses Transportprotokoll für Pakete

➤ Hauptfunktion:

– Demultiplexing von Paketen aus der Vermittlungsschicht

➤ Zusätzlich (optional):

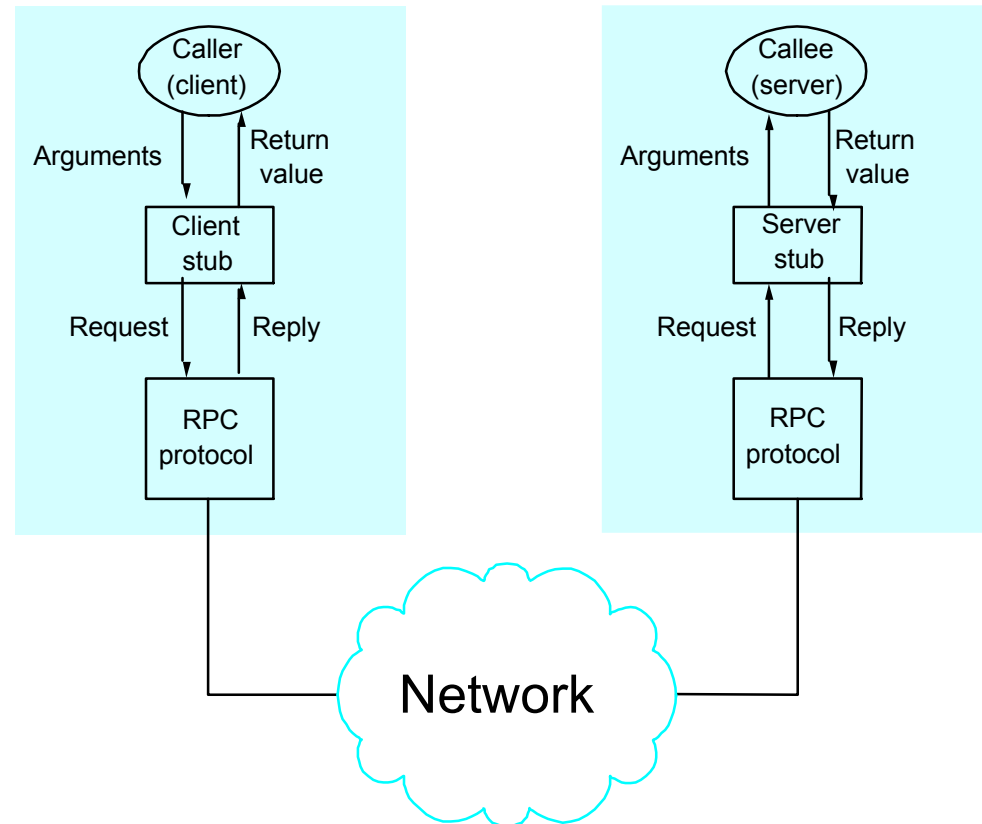
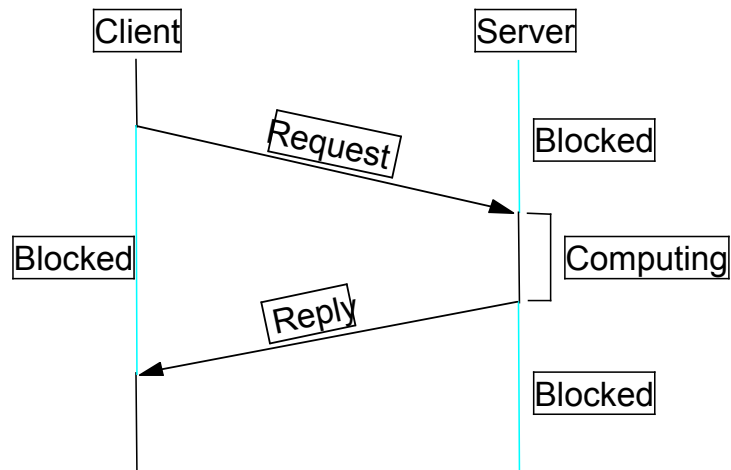
– Checksum aus UDP Header + Daten





Remote procedure call – Struktur

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



- **Für komplexere Interaktion:**
 - Aufruf einer Funktion auf einen anderen Rechner
- **Ziel: Transparentes Protokoll für den Aufrufer/Aufgerufenen**

Ende der 19. Vorlesung



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

Systeme II
Christian Schindelhauer
schindel@informatik.uni-freiburg.de